PATH PLANNING FOR SPATIAL ROBOTS WITH MULTIPLE SPHERICAL AND CYLINDRICAL OBSTACLES INSIDE THE WORKSPACE

BY

MENQ-DAR SHIEH

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1990

With love to Dad, Mom, and my beloved wife Bih-Jau Chiou

ACKNOWLEDGEMENTS

The author would like to express his sincere appreciation to his supervisory committee chairman, Dr. Joseph Duffy, for his invaluable guidance, support, and inspiration throughout the course of this research. Special thanks are due to the committee professors, Dr. Carl D. Crane, Dr. Ali Seireg, Dr. John Staudhammer, and Dr. Ralph G. Selfridge, for their guidance and comments on this work.

He would like to thank his fellow students in CIMAR (Center for Intelligent Machines and Robotics) for their friendship and sharing of knowledge for the last several years. Special thanks should be extended to Paul Lewicki and Jau-Liang Chen for their help in this dissertation.

Finally, the author sincerely thanks his family and his beloved wife, Bih-Jau Chiou, for their spiritual encouragement and endless love.

TABLE OF CONTENTS

page

ACKNOWLEDGEMENTS iii

ABSTRACT vi

CHAPTERS

1. INTRODUCTION 1

 1.1 Review of Literature 1

 1.2 Scope of Study 10

2. AUTONOMOUS PATH PLANNING FOR A SPATIAL 4R MANIPULATOR WITH A SINGLE SPHERICAL OBSTACLE INSIDE THE WORKSPACE 16

 2.1 Introduction 16

 2.2 Determination of a Truncated Pyramid 17
  2.2.1 Derivation of *NA* of the End Effector Tip p_4 18
  2.2.2 Determination of a Bounding Trapezoid .. 21
  2.2.3 Truncated Pyramid 23

 2.3 Determination of a Right Circular Torus 23

 2.4 Path Planning 24
  2.4.1 Path Planning for Avoiding a Truncated Pyramid 24
  2.4.2 Path Planning for Avoiding a Right Circular Torus 25
  2.4.3 Path Planning for Avoiding a Truncated Pyramid and a Right Circular Torus 27

3. AUTONOMOUS PATH PLANNING FOR A SPATIAL 4R MANIPULATOR WITH MULTIPLE SPHERICAL OBSTACLES INSIDE THE WORKSPACE 40

 3.1 Introduction 40

3.2 Determination of Collision Free Paths with Multiple Spherical Obstacles inside the Workspace (Method I) 41
3.2.1 Transformation of a Truncated Pyramid into a Rectangle in the *IACS* 41
3.2.2 Determination of Free Paths in the *IACS* . 43
3.2.3 Determination of the Inner and Outer Boundaries of the End Effector Tip 47
3.2.4 Determination of Collision Free Paths in the Cartesian Coordinate System 49
3.2.5 Simulation for the Path Planning Algorithm (Method I) 51

3.3 Determination of Collision Free Paths with Multiple Spherical Obstacles inside the Workspace (Method II) 52
3.3.1 Determination of Group I and Group II Spherical Obstacles 52
3.3.2 Path Planning for Multiple Spherical Obstacles (Method II) 53

4. AUTONOMOUS PATH PLANNING FOR A SPATIAL 4R MANIPULATOR WITH MULTIPLE CYLINDRICAL OBSTACLES INSIDE THE WORKSPACE AND A COMPUTER GRAPHICS SIMULATION ON A $T^3$586 ROBOT 73

4.1 Introduction 73

4.2 Determination of Collision Free Paths with Multiple Cylindrical Obstacles inside the Workspace 74
4.2.1 Representation of Cylindrical Obstacles . 74
4.2.2 Path Planning for Multiple Cylindrical Obstacles 75

4.3 Computer Graphics Simulation on $T^3$586 Robot .. 80

5. CONCLUSIONS AND FUTURE WORK 95

REFERENCES 99

BIOGRAPHICAL SKETCH 105

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

PATH PLANNING FOR SPATIAL ROBOTS WITH MULTIPLE SPHERICAL AND CYLINDRICAL OBSTACLES INSIDE THE WORKSPACE

By

MENQ-DAR SHIEH

December, 1990

Chairman: Dr. Joseph Duffy
Major Department: Mechanical Engineering

Algorithms which can rapidly generate collision free paths for the end effector tip of a spatial 4R manipulator with multiple spherical or cylindrical obstacles inside the workspace have been successfully developed. The spatial 4R manipulator consists of a vertical joint attached to the first joint of a planar 3R robot whose three revolute joints are parallel to the ground.

The algorithms are based on the geometry of the manipulator workspace. Truncated pyramids and a right circular torus are used to contain the non-reachable workspaces of the end effector tip with respect to spherical obstacles and a central void. Further, the truncated pyramids are transformed into rectangles in the *Inclination Angle Coordinate System* (*IACS*). In this way, the problem of guiding the spatial

4R manipulator while avoiding spherical obstacles is reduced to that of moving a point and at the same time avoiding rectangles in the *IACS*. The complexity of the path planning is reduced from the 3D case to the 2D case in the *IACS*, and the speed of generating the collision free paths is improved significantly without losing the characteristics of 3D path planning. The free paths in the *IACS* are, then, transformed into the 3D collision free paths in the Cartesian coordinate system. It is important to note that the collisions between the links of the manipulator and the spherical obstacles are already considered as well as the determination of the collision free paths of the end effector tip.

The algorithms are further applied to cylindrical type of obstacles, which are not necessarily fixed to the ground or ceiling, and are neither parallel nor perpendicular to the ground. This provides a practical way for collision free path planning with floating pipes inside the workspace.

The algorithms have been successfully implemented in a Silicon Graphics 4D-70GT workstation to verify the results. The computation time for generating 10 collision free paths with 7 spherical obstacles or 10 cylindrical obstacles inside the workspace is 1 or 2 seconds.

Also, the algorithms, which are designed as interactive programs, are modified to guide a spatial T^{3}586 robot around pipes with circular cross sections.

CHAPTER 1
INTRODUCTION

1.1 Review of Literature

It is very important in industrial automation that robots perform tasks while avoiding obstacles strewn in working environments. The information on working environments may be known at the outset, so robots can perform tasks according to predetermined instructions generated from implemented algorithms, most of which are developed emphasizing optimal trajectory planning. However, if the information about working environments is not available, it is necessary to gather some information of the working environments with the help of different types of sensors and then use the information to generate trajectories which can guide robots to perform pre-specified tasks. Rather than spending much computing time to find the optimal trajectory, it is preferable to develop algorithms which reduce considerably the computation time for generating collision free trajectories such that robots can perform tasks while avoiding obstacles inside the working areas.

Recently, researchers who work on collision free path planning of manipulators in an environment with obstacles present endeavor to develop algorithms which have simplicity, high accuracy, small memory size and short execution time. It is however difficult to produce algorithms which include all these features.

Udupa [1] proposed a solution, which was regarded as the first work on obstacle avoidance for a computer controlled spatial manipulator, for planning safe trajectories for a manipulator with two movable links and multiple degrees of freedom. When the goal position and orientation of the hand were given, the system determined a complete trajectory that enabled the manipulator to maneuver safely into the goal configuration. A hierarchical decomposition was used to reduce the complexity of the Planning Problem. Boom Planning was treated as planning trajectories in empty space, and forearm planning was treated as a collision avoidance problem.

Lozano-Perez [2-5] derived *configuration space* (C-space), which was used to plan collision free motions for kinematically simple and general manipulators with revolute joints. The configuration-space obstacles were derived for an n degree-of-freedom manipulator, which were approximated by lists of n-1 dimensional slices. The complement of the C-space obstacles was called the free space. A* search was introduced to track a collision free path through a list of regions connecting the initial point to the goal. The algorithm had several advantages: it was simple to implement, it was fast for manipulators with few degrees of freedom, and it could deal with cluttered environments and non-convex polyhedral obstacles. However, the disadvantages were the loss of accuracy, the increase of storage, and the loss of efficiency for the robot with many joints.

Khatib [6] used an *artificial potential field* concept for the real-time obstacle avoidance for manipulators and mobile robots. He demonstrated an effective component of low-level real-time control to solve the collision avoidance problems which were generally treated as high level planning. This method had been extended to moving obstacles by using visual sensing and a time-varying artificial potential field and had been implemented in the COSMOS system for a PUMA 560 robot.

Lumelsky [7] developed an algorithm to plan collision free paths for a planar arm with revolute joints in an environment with obstacles. Since the environment was considered to be unknown, feedback information about its intermediate surroundings was sent back to the system in order to generate reasonable paths reaching the goal position. Therefore, the local algorithm was used to compute the path one step at a time. The approach was referred to as the Continuous Path Planning because the local algorithm generated paths continuously based on the incoming information. The motion planning problem was reduced to the analysis of simple closed curves on the surface of an appropriate two-dimensional manifold. Further, Cheung and Lumelsky [8] considered the problem of sensor-based path planning for planar and simple three-dimensional robot arms operating among unknown obstacles of arbitrary shape. Whenever an obstacle was present, the robot arm attempted to slide along the local tangent of the obstacle. They also addressed the hardware and lower-level control issues related to the Dynamic Path Planning (DPP) and, specifically, the problem of instrumenting a robot arm with sensitive skin and enabling it with the capability to interpret the sensory data generated by the skin for the purposes of obstacle avoidance. Lumelsky and Sun [9] presented a unified methodology for designing DDP algorithms for two- and three-dimensional two-link arm manipulators. The approach is independent of the specifics of the shape of the arm links or obstacles in the environment.

Young and Duffy [10,11] presented a motion planning procedure for planar 3R manipulators which provided the robots with abilities to articulate and, hence, to avoid interference with other robots or with obstacles. The relative locations of a pair of links were defined by a series of parallelograms which were mapped onto the same rectangle on the X_1X_2 plane with different values in the X_3 axis (see Fig. 1.1); thus, these rectangles form a rectangular prism. An allowable trajectory manifold was determined to calculate the orientation range of link a_{23}. The orientation range of link a_{23} was used to determine the corresponding ranges of orientation of links a_{34} and a_{12} such that all three links avoided interference. Choi et al. [12] extended the idea to three dimensions in order to develop an articulation theory for spatial PUMA and TRS robots. The geometric capability of a manipulator was analyzed using the workspace and dexterity of the manipulator. The permissible orientations for a regional structure of a robot were determined under given geometric constraints. Further, the relative displacements of an object with respect to a reference object were mapped into image points using an affine transformation. An inverse mapping was used as a rapid decision-making tool for collision free path planning, which reduced the computation time significantly. The application of the theory to a collision free path planning problem was demonstrated using a PUMA robot.

Tseng [13] presented a trajectory planning to rapidly generate a collision free path for a Cincinnati Milacron T^3-776 manipulator. The obstacles were assumed to be polyhedral objects with vertical sides and flat top surfaces, and their positions and dimensions were known. The links of the manipulator were treated as straight line segments by using the concept of obstacle expansion. A reachable end effector point was determined on each vertical plane through the Z axis and each vertical side of the expanded obstacle. An Archimedes's spiral was chosen to modify the path through the initial position, the reachable end effector points, and the final goal. The advantage is its computation speed for generating collision free paths. However, the obstacles were constrained to be fixed to ground.

Here, it is important to note that the obstacles used by Lozano-Perez [4,5], Choi et al. [12], and Tseng [13] were constrained to be polyhedral objects with vertical sides and flat top surfaces. These constraints were removed in the work done by Shieh and Duffy [14], where a spherical obstacle was introduced and a path planning algorithm for the end effector tip of a spatial manipulator was developed. The algorithm was further extended (see chapters 3 and 4) to the cases with cylindrical obstacles with no constraints on the orientations.

Shiller and Dubowsky [15] presented a global time optimal motion planning of robotic manipulators with obstacles in their environment. The examples showed that only the motions of the first three joints were optimized, while the orientation of the end-effector remained fixed. Nonlinear manipulator dynamics, actuator constraints, joint limits, and obstacles were all taken into account. Graph search and hierarchical pruning techniques were used for a global path search over the manipulator workspace which consisted of a 10 * 10 * 10 grid. The computation time for the hierarchical pruning was about 3 minutes of the CPU time, for the local optimization was about 10 minutes, and for the 3-D cases was in the order of 2 hours on a microcomputer, μ VAX II. They [16] further presented a global time optimal motion planning of robotic manipulators with obstacles, actuator, gripper, and payload constraints. The method was computationally practical and had been implemented for the optimal trajectory planning of general six-degree-of-freedom manipulators. However, it was an off-line planning technique, the computation time, for example, for an optimization for a six-DOF manipulator with five to nine control points was completed within 10 to 20 minutes of CPU time. The algorithm also needs an initial path that avoids obstacles. The initial path was determined with the help of a computer graphics technique. However, in a cluttered environment, it was difficult to select an obstacle-free path and it might not be clear which initial path would converge to yield a better solution.

Chen and Vidyasagar [17] used the joint space approach to model the obstacles, called the J*obstacles, which were enclosed by an algebraic closed curve (for instance, an ellipse) in the joint space. These expressions were then used to formulate the optimal trajectory planning problem as an optimal control problem in constrained state space. Meyer and Benedict [18] also exploited geometric characteristics of configuration space obstacles to path planning.

Khosla and Volpe [19] presented a new obstacle avoidance potential based on superquadrics, which is viable for a much larger class of object shapes. The repulsive forces exerted on links were calculated based on the shortest distance to an object, which was used to avoid collisions with obstacles.

A simulation of two and three link planar manipulators interacting with an artificial potential was shown. A similar work was also done by Hwang and Ahuja [20].

Xia, Jiang, and Lii [21] focused on how to reduce memory size and running time. They used intuitive decision to pass the obstacle sideways or above as the first level optimal path planning. The second level optimal path planning was used in the regions containing obstacles. The free space was described in detail by means of an octree, and a precise obstacle avoidance path was searched by using the graph of the neighbors of the octree.

Muck [22] used a concept called the Constraint Space Method in motion planning. The method was based on the concept of grouping points with identical constraint of motion together to form regions. Collision free paths can then be determined by connecting regions that share common boundaries. The algorithm was applied to single body and articulated manipulator motion planning.

Chen and Duffy [23,24] had comprehensively discussed the parallel operation of a pair of robots. Zaharakis and Guez [25] determined a time optimal obstacle avoidance trajectory for a robot of known dynamic capability where obstacles were rectangular with random size and position in the workspace. Chang et al. [26] presented an on-line Cartesian path trajectory planning algorithm for Robot Manipulators, which generated approximate joint paths for any curve in Cartesian space. Chien et al. [27] proposed an on-line algorithm to generate collision free trajectories for multiple robot manipulators controlled independently by means of PD-controllers. Eltimsahy and Yang [28] developed a near minimum-time suboptimal controller for a robot manipulator to follow a prespecified path and to avoid obstacles in the workspace.

Although the problem of navigation of a mobile robot is much different from that of path planning for a manipulator, it is worth having the knowledge for the work relevant to the motion planning for mobile robot. Kant and Zucker [29], and Fujimura and Samet [30] proposed algorithms to plan collision free trajectories for a mobile robot in the plane among stationary and moving obstacles. Avnaim et al. [31] developed a two dimensional motion planning algorithm for polygonal objects amidst polygonal obstacles. Rao et al. [32] presented an algorithm to navigate a point robot, which was equipped with a "see from distance" type of sensor, through a sequence of destination points amidst unknown stationary polygonal obstacles in a two dimensional terrain. Kanayama [33] proposed a new idea of planning safer paths for robot motion using algebraic cost functions, which can be applied to path design both for robot manipulators and autonomous vehicles.

Other work related to collision free path planning is presented in [34-41]. References [42-44] emphasize control schemes.

1.2 Scope of Study

Algorithms have been successfully developed using the geometry of a manipulator workspace to rapidly generate collision free paths for the end effector tip of a spatial 4R robot (see Fig. 2.1) with multiple spherical obstacles or cylindrical obstacles inside the workspace. Since the end effector of a general spatial robot manipulator with 6 degree of freedom is small compared with the sizes of upper arm and forearm, it is assumed in this work that the end effector of the robot moves with constant orientation during the entire motion. The orientation of the end effector is measured relative to the horizontal of the vertical plane on which the robot lies.

This research stems from the work done by Lipkin, Torfason, and Duffy [45], who developed time efficient algorithms for determining the possibilities of a rectilinear motion of the end effector of a planar 3R manipulator. The algorithms were based upon a geometrical quantification of the manipulator dexterity and workspace.

The algorithms were extended by Shieh and Duffy [46,47] by placing a single circular obstacle inside the workspace. The geometry of the reachable area and non-reachable area of the end effector tip was intensively studied. The reachable area of the end effector tip facilitates the determination of motion capability simply because it enables one to check whether the target position lies within the reachable area or not. Therefore, the reachable area was used for determining a collision free path for the manipulator to move around the circular obstacle. A collision free path consisting of a sequence of line segments was generated autonomously within 0.2 seconds. The computation time mentioned throughout this work is based on the calculation speed of the Silicon Graphics 4D-70GT workstation.

The algorithms were further extended by Shieh and Duffy [48] with multiple circular obstacles inside the workspace. A computer graphics technique was used to determine the reachable area of the end effector tip. This was done by finding the individual reachable area with respect to each of the circular obstacles, and then finding a common area of all the individual reachable areas using a graphics technique to superimpose all the images of the reachable areas. After the determination of the reachable area for the multiple circular obstacles, another computer graphics technique called the "color recognition" technique was used to determine the intermediate points within the reachable area. Then, a collision free path was generated from a given initial position, through the intermediate positions, and then to a final target position. The computation time for generating autonomously a collision free path for the manipulator was within 1 second.

With the knowledge of the 2D path planning, a 3D path planning has been again intensively studied by Shieh and Duffy [14], where the manipulator is a spatial type 4R manipulator, the obstacles are of a spherical type, and the reachable and non-reachable areas become reachable and non-reachable workspaces. For a spatial 4R manipulator with multiple spherical obstacles inside the workspace, there are two types of non-reachable workspaces of the end effector tip. The first one is the non-reachable workspace which cannot be reached by the end effector tip because of the existence of a spherical obstacle. The computation of the boundaries of this type of non-reachable workspace is complicated, and the workspace is considered to be bounded by a relatively simple truncated pyramid. The second type of non-reachable workspace is a central void. This is caused by the link dimensions of the manipulator, and this type of non-reachable workspace can be represented by a right circular torus.

With the representations of the truncated pyramid and the right circular torus, the problem of guiding a spatial 4R manipulator while avoiding spherical obstacles is reduced to that of moving a point (tip of the end effector) and at the same time avoiding truncated pyramids and/or a right circular torus. It is important to note that the collisions between the links of the manipulator and the spherical obstacles have already been taken into consideration, even though the collision free paths are determined only for the end effector tip.

Further, the truncated pyramids are transformed into rectangles in the *Inclination Angle Coordinate System* (*IACS*). The problem of guiding the manipulator to avoid spherical obstacles is further reduced to that of moving a point and at the same time avoiding rectangles in the *IACS*. In this way, the complexity of the path planning is reduced from a 3D case down to a 2D case in the *IACS* such that the speed of generating the collision free paths is improved significantly. The free paths determined in the *IACS* are then transformed into the final 3D collision free paths in the Cartesian coordinate system. This approach produces a practical way for determining collision free paths without losing the characteristics of 3D path planning.

The algorithms are further applied to cylindrical type of obstacle which can be modelled by a series of spherical obstacles with their centers lying on the central axis of the cylinder. Furthermore, the cylindrical obstacles are not necessarily fixed to the ground or ceiling, and are neither parallel nor perpendicular to the ground. This provides a method for rapid generation of collision free path with floating pipes inside the workspace.

The algorithms have been implemented in the Silicon Graphics 4D-70GT workstation to verify the results. The computation time for generating 10 collision free paths with 7 spherical obstacles or 10 cylindrical obstacles inside the workspace is 1 or 2 seconds.

Also, the algorithms are modified to guide a spatial $T^3$586 robot around pipes with circular cross sections. Since the spatial $T^3$586 robot has real dimensions for each link, it is assumed that the robot's links are bounded by cylinders. The radii of spherical obstacles are enlarged according to the diameters of the bounding cylinders, and the robot's links are shrunk into line segments. In this way, the algorithms developed above can be used for determining the collision free paths for the spatial $T^3$586 robot.

In order to permit a user to verify results easily, the computer graphics simulation is designed as an interactive program which enables a user to change arbitrarily the number, locations and sizes of the spherical or cylindrical obstacles, and the initial and target positions of the end effector tip. Likewise, the orientation of the end effector can be specified arbitrarily by using the mouse device.

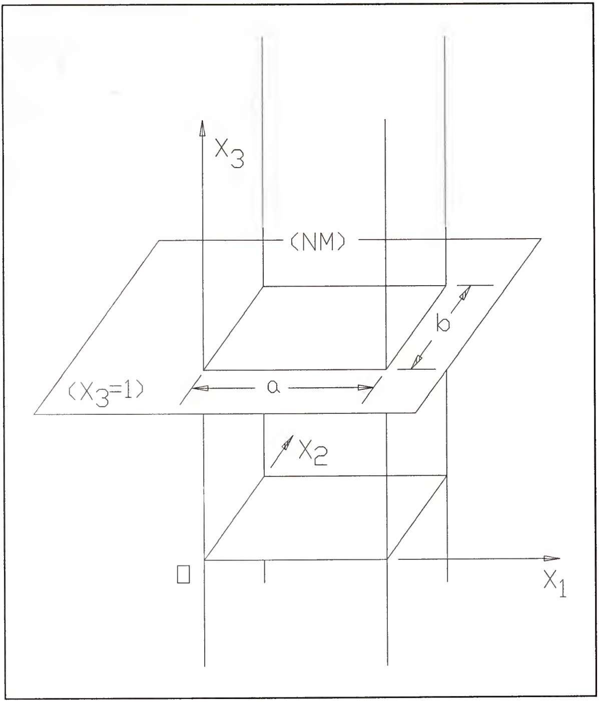


Figure 1.1 Mapping space.

CHAPTER 2
AUTONOMOUS PATH PLANNING FOR A SPATIAL 4R MANIPULATOR WITH A SINGLE SPHERICAL OBSTACLE INSIDE THE WORKSPACE

2.1 Introduction

An algorithm has been developed to rapidly generate a collision free path for the end effector tip of the spatial 4R manipulator with a single spherical obstacle inside the workspace. Since the end effector of a general spatial robot manipulator with 6 degree of freedom is small compared with the sizes of upper arm and forearm, it is assumed in this work that the end effector of the robot moves with constant orientation along a series of straight lines. The orientation of the end effector is measured relative to the horizontal of the vertical plane on which the robot lies. The spatial 4R manipulator is shown in Fig. 2.1, which has a vertical joint attached to the first joint of a planar 3R robot whose three revolute joints are parallel to the ground.

In addition to the workspace outside the maximum reachable range of the robot, there are two non-reachable workspaces for the end effector tip. One is the non-reachable workspace which cannot be reached by the end effector tip because of the existence of the spherical obstacle. The computation of the boundaries of this non-reachable workspace is complicated, and it is considered to be bounded by a relatively simple truncated pyramid.

The other non-reachable workspace is a central void. This is caused by the link dimensions of the robot, and it can be represented by a right circular torus. The problem of guiding a spatial 4R manipulator while avoiding a spherical obstacle is reduced to that of moving a point (tip of the end effector) and at the same time avoiding a truncated pyramid and/or a right circular torus.

In this way, the computation time of the path planning algorithm is reduced, and the speed of generating a collision free path is improved significantly.

2.2 Determination of a Truncated Pyramid

A truncated pyramid is determined according to a trapezoid which bounds the Non-reachable Area (*NA*) of the end effector tip on a vertical plane drawn through the center of the spherical obstacle and the robot center $\underline{O}$. The robot center $\underline{O}$ is defined as the point at the intersection of the axes of the first and second joints (see Fig. 2.1). This vertical plane cuts the sphere in a great circle and the geometry for deriving the *NA* of the end effector tip on the vertical plane [1] depends on the distance from the center of

[1] It is assumed throughout this analysis that the robot does not change its configuration viz. the algorithm is developed here for the robot in the elbow down position. A similar algorithm can be easily developed for the elbow up position.

the great circle to the robot center, the dimension of the links, and the radius of the great circle. There are essentially two distinct cases which must be considered.

Case (1): The great circle lies outside the boundary of joint 2 (the intersection of the link a_{12} and link a_{23}) and lies totally or partially inside the outer boundary of the end effector tip (see Fig. 2.2(a)).

Case (2): The great circle lies inside or intersect the boundary of joint 2 (see Fig. 2.2(c)).

2.2.1 Derivation of *NA* of the End Effector Tip p4

Because it is assumed that the end effector moves with constant orientation, the *NA* of the end effector tip can be determined by firstly determining the *NA* for the wrist point p_3 (see Fig. 2.2(a)) and then translating it parallel to the vector $\underline{a}_{34}$ through a distance a_{34} (the length of the end effector). The vector $\underline{a}_{34}$ measures the constant of orientation of the end effector in the vertical plane.

Case (1): The wrist point p_3 cannot move into two shaded areas *NA*(a) and *NA*(m) (see Fig. 2.2(a)). The *NA*(a) of the wrist, which is defined by the central limit circle C_{3i} of p_3, will be discussed in section 2.3. The *NA*(m) is the area

These pair of algorithms form a basis for developing a further algorithm which makes use of the change of configuration to avoid obstacles.

bounded by the coupler curves $D_1D_2D_3D_4$ and $E_1E_2E_3E_4$ generated by point p_3, the circular arc D_1E_1 and the great circle C_{ob}. The derivation of these pair coupler curves was discussed by Shieh and Duffy [46] and will not be repeated in detail here. Briefly, for example, the coupler curve $D_1D_2D_3D_4$ can be determined as follows. Suppose that joint 2 (point p_2) is initially coincident with point B_1 and link a_{23} (line segment p_2p_3) is tangent to the obstacle. As joint 2 moves from B_1 to B_4 with a_{23} maintaining tangency, the wrist (point p_3) will trace a coupler curve from D_1 to D_4. This motion can be modeled by a planar RRRP mechanism (Duffy [49]). Points D_1, D_4 and two other intermediate points D_2 and D_3 are introduced and the coupler curve is approximated by the sequence of line segments D_1D_2, D_2D_3, and D_3D_4. The points D_2 and D_3 are determined by choosing the points B_2 and B_3 on the arc B_1B_4, where $B_1B_3=B_3B_4$ and $B_1B_2=B_2B_3$. Point p_3 is coincident with points D_2 and D_3, when joint 2 is coincident with points B_2 and B_3 and the link a_{23} is tangent to the obstacle.

The *NA* of the end effector tip is now determined simply by translating the *NA*(m) of the wrist point p_3 parallel to the direction $\underline{a}_{34}$ through a distance a_{34}. This is labeled $D_1'D_2'D_3'D_4'E_4'E_3'E_2'E_1'$ in Fig. 2.2(b). In addition to this, there is a further non-reachable area, *NA*(u), which is bounded by the great circle (C_{ob}), the translated image (C_{ob}') of the great circle, and two common tangent lines (L_3 and L_4)

to both the circular obstacle and the virtual circular obstacle.

The *NA*(u) together with the translated image of the *NA*(m) of the wrist define the *NA* of the end effector tip for this case.

<u>Case (2)</u>: The wrist point p_3 cannot move into two shaded areas *NA*(a) and *NA*(s) (see Fig. 2.2(c)). The *NA*(a) of the wrist will be discussed in section 2.3. The *NA*(s) is the area bounded by the circular arcs D_1D_2 and E_1E_2 generated by wrist point p_3, the circular arcs D_1E_1 and D_2E_2. The derivation for these pair of circular arcs D_1D_2 and E_1E_2 was discussed by Shieh and Duffy [46] and will not be repeated in detail here. Briefly, for example, the circular arc D_1D_2 can be determined as follows. Given that the wrist point p_3 is initially coincident with point D_1 and link a_{12} (line segment p_1p_2) is tangent to the obstacle, also it is assumed that joint 2 (point p_2) can rotate from 0 to 180 degree (elbow down configuration), then the wrist point p_3 will trace a circular arc from D_1 to D_2.

The *NA* of the end effector tip is now determined simply by translating the *NA*(s) of the wrist point p_3 parallel to the direction $\underline{a}_{34}$ through a distance a_{34}. This is labeled $D_1'D_2'E_2'E_1'$ in Fig. 2.2(d). In addition to this, there is a further non-reachable area, *NA*(u), which has been discussed in case (1).

The *NA*(u) together with the translated image of the *NA*(s) of the wrist define the *NA* of the end effector tip for this case.

2.2.2 Determination of a Bounding Trapezoid

A trapezoid is now derived to bound the rather complicated boundary of the *NA* for the end effector tip determined above. It is simple and efficient to use a trapezoid to bound the *NA* for the end effector tip but there is loss of accuracy because of the ignorance of some reachable areas, such as the non-shaded areas inside the trapezoid and the outer boundary C_{30}' (see Fig. 2.2(b)). Fortunately, in most cases, it is enough to determine collision free paths by using the bounding trapezoid approximation. In case the end effector tip p_4 lies inside the reachable areas (which are ignored because of using the bounding trapezoid method), then the reachable areas become very important in the path generation algorithm. The improvement of the algorithm for this case will be discussed in detail in section 4.2. Here, the bounding trapezoid is determined for general path planning cases.

Case (1): The great circle lies outside the boundary of joint 2 (point p_2) and lies totally or partially inside the outer boundary of the end effector tip. The bounding trapezoid can be determined as follows. In Fig. 2.2(b), lines L_5 and L_6 are drawn from the center $\underline{O}$ of the manipulator. L_5 is tangent to the great circle C_{ob} and L_6 passes through point E_3'. Further, a line N_1 is drawn tangent to the great circle C_{ob} and perpendicular to the bisector of the angle θ_1. Finally, the line N_2 is drawn parallel to N_1 and tangent to the circle C_{3O}', which is the outer boundary of the end effector tip. These four lines form the trapezoid which bounds the *NA* of the end effector tip.

Case (2): The great circle lies inside or intersect the boundary of joint 2. The bounding trapezoid can be determined by drawing lines L_5 and L_6 from the center $\underline{O}$ of the manipulator (see Fig. 2.2(d)). L_5 is tangent to the great circle C_{ob} and L_6 is tangent to the circular arc E_1E_2. Further, a line N_1 is drawn from point A (or point E, depends on which point is closer to the robot center $\underline{O}$) and perpendicular to the bisector of the angle θ_1. The points A and E are the intersections of the lines L_5 and L_6 with the circle $C_{3i'}$ near the *NA*. Finally, the line N_2 is drawn parallel to N_1 and tangent to the circle C_{3O}', which is the outer boundary of the end effector tip. These four lines form the trapezoid which bounds the *NA* of the end effector tip.

2.2.3 Truncated Pyramid

For spatial motion, a truncated pyramid is determined from the planar bounding which contains the entire non-reachable workspace of the end effector p_4 due to the spherical obstacle.

This truncated pyramid is determined by firstly choosing four planes (π_1, π_2, π_3 and π_4) passing through the four edges of the trapezoid T_{ABDE} and perpendicular to the vertical plane π_T where the trapezoid lies (see Fig. 2.3). Two other planes (π_5 and π_6) are chosen which pass through the Z axis and are tangent to the sides of the spherical obstacle. These six planes determine the truncated pyramid.

2.3 Determination of a Right Circular Torus

The right circular torus is the non-reachable workspace where the end effector tip cannot reach due to the restriction of link dimensions. It is derived, firstly, by determining the central limit circle C_{3i} (inner boundary of the wrist point p_3) on the vertical plane passing through the center of the spherical obstacle and the robot center $\underline{O}$. Then, the central limit circle C_{3i} is translated along the direction ($\underline{a}_{34}$) of the end effector through a distance a_{34}. This determines the NA(a) of the end effector tip in the vertical plane (see Fig. 2.4).

Rotating the NA(a) through 2π about the Z axis generated a right circular torus, which is the non-reachable workspace of the end effector tip.

2.4 Path Planning

In this section, the possibility of rectilinear motion of the end effector tip from an initial position p to a final position q is determined. If the line segment pq does not intersect either the truncated pyramid or the right circular torus, then rectilinear motion is possible. However, if the line segment pq intersects either the truncated pyramid or the right circular torus, then rectilinear motion is not possible, and a number of intermediate points can be chosen such that the end effector of the manipulator can follow a sequence of line segment to reach the final goal position.

2.4.1 Path Planning for Avoiding a Truncated Pyramid

If the line segment pq intersects the truncated pyramid (see Fig. 2.5), rectilinear motion of the end effector tip is not possible. In this case, a vertical plane κ_1 through points p and q is drawn. The vertical plane κ_1 is enough to determine the collision free paths for most cases with a single spherical obstacle inside the workspace. However, a modified algorithm is developed in chapter 3 to generate collision free paths for multiple spherical obstacles inside the workspace. The vertical plane κ_1 intersects the edges of the truncated pyramid in four points E_1, E_2, E_3 and E_4. There are two possible collision free paths from p to q, one is pE_1E_2q and the other is pE_4E_3q. The shorter of these two paths is chosen as the collision free path.

2.4.2 Path Planning for Avoiding a Right Circular Torus

The equation for the right circular torus is obtained by rotating C_{3i}' about the Z axis (see Fig. 2.4 and also Hunt [50]).

$$\{ (x^2 + y^2 + (z-c)^2) - (a^2 + b^2) \}^2 = 4 a^2 \{ b^2 - (z-c)^2 \}, \quad (2.1)$$

where $a = a_{34y}$, $b = a_{12} - a_{23}$, $c = a_{34z}$.

A point (x, y, z) which lies on the line segment pq can be expressed by

$$\begin{aligned} x &= p_x + \lambda (q_x - p_x), \\ y &= p_y + \lambda (q_y - p_y), \\ z &= p_z + \lambda (q_z - p_z), \end{aligned} \quad (2.2)$$

where $0 \leq \lambda \leq 1$.

Substituting (x, y, z) into the equation of the right circular torus, and regrouping terms yields a fourth degree of polynomial in λ which can be expressed in the form,

$$F(\lambda) = A_1 \lambda^4 + A_2 \lambda^3 + A_3 \lambda^2 + A_4 \lambda + A_5 = 0, \qquad (2.3)$$

where

$$A_1 = k_1^2,$$
$$A_2 = 2k_1k_2,$$
$$A_3 = k_2^2 + 2k_1k_3 - k_4,$$
$$A_4 = 2k_2k_3 - k_5,$$
$$A_5 = k_3^2 - K_6.$$

Further,

$$k_1 = (q_x-p_x)^2 + (q_y-p_y)^2 + (q_z-p_z)^2,$$
$$k_2 = 2[\ p_x(q_x-p_x) + p_y(q_y-p_y) + (p_z-c)(q_z-p_z)\ ],$$
$$k_3 = p_x^2 + p_y^2 + (p_z-c)^2 - a^2 - b^2,$$
$$k_4 = -4a^2\ (q_z-p_z)^2,$$
$$k_5 = -8a^2\ (p_z-c)(q_z-p_z),$$
$$k_6 = 4a^2\ (b^2 - (p_z-c)^2),$$

If there are real roots for $F(\lambda)=0$ and $0 \le \lambda \le 1$, then the line segment pq intersects the right circular torus and rectilinear motion from p to q is not possible. An alternative path will now be sought.

A right circular cylinder with axis Z is drawn which contains the entire right circular torus (see Fig. 2.6). The radius of the cylinder is $a_{34y}+(a_{12}-a_{23})$ and the length is $2(a_{12}-a_{23})$, where $h_1 = a_{34z}+(a_{12}-a_{23})$ and $h_2 = a_{34z}-(a_{12}-a_{23})$. A vertical plane κ_1 through points p and q is chosen and the plane intersects the cylinder in a rectangle with vertices E_6, E_7, E_8 and E_9. This yields two possible collision free paths. One is pE_6q and the other one pE_9E_8q. The shorter of these two paths, i.e., pE_6q, is chosen as the collision free path and the point E_6 is thus the intermediate point for this alternative path.

2.4.3 Path Planning for Avoiding a Truncated Pyramid and a Right Circular Torus

If the line segment pq intersects both the truncated pyramid and the right circular torus, rectilinear motion of the end effector from p to q is not possible. Hence, it is required to produce a path searching algorithm to determine a collision free path from p to q. It is important to note that the path searching algorithm can be extended to the case where multiple spherical obstacles lie inside the workspace.

Here, a vertical plane κ_1 through points p and q is chosen, and the vertical plane intersects the truncated pyramid and the right torus into two polygons $E_1E_2E_3E_4E_5$ and $E_6E_7E_8E_9$ (see Fig. 2.7). The coordinates of each of the vertices of the two polygons are determined to establish whether they lie inside or outside the workspace of the end effector. If all the vertices E_1, E_2, E_3, E_4, E_5 are imaginary, then the spherical obstacle is so large that motion form p to q is not possible and the algorithm will stop. Clearly, if any vertex lies outside the workspace, it can not be considered as an intermediate point to determine a free path.

A searching procedure starts from the initial position p to check all the vertices of the two polygons. If the vertex can be reached by the end effector from the initial position p in a straight line motion, the second searching procedure will start from this vertex to the next reachable vertex. The same procedure is repeated until the final position q is reached.

After the search procedure, there are up to a maximum of seven collision free paths, which are pE_1E_2q, pE_9E_6q, $pE_5E_4E_3q$, $pE_5E_4E_6q$, $pE_9E_4E_3q$, $pE_9E_4E_6q$ and $pE_9E_8E_7q$. For example (see Fig. 2.7), the shortest is $pE_5E_4E_6q$, and it is chosen as the free path. Figure 2.8 shows the entire process of the determination of the seven collision free paths. The 'X' in the figure 2.8 indicates the path generation is terminated to avoid a point being repeated in a free path.

The algorithm can be extended to the case of multiple spherical obstacles inside the workspace.

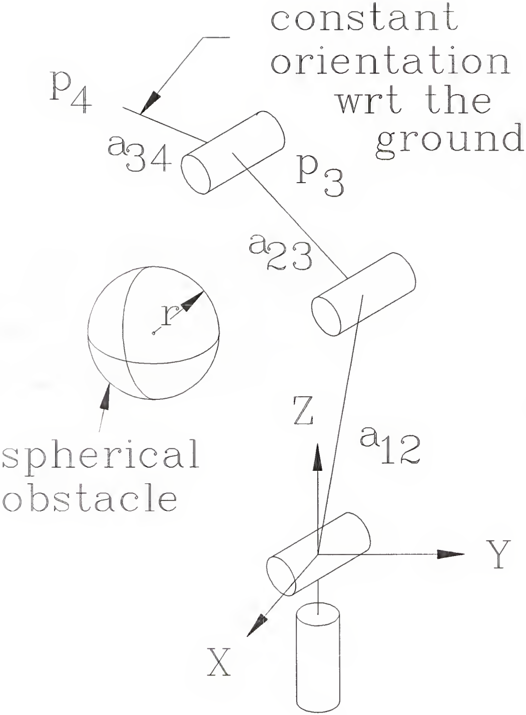


Figure 2.1 A spatial 4R manipulator and a spherical obstacle.

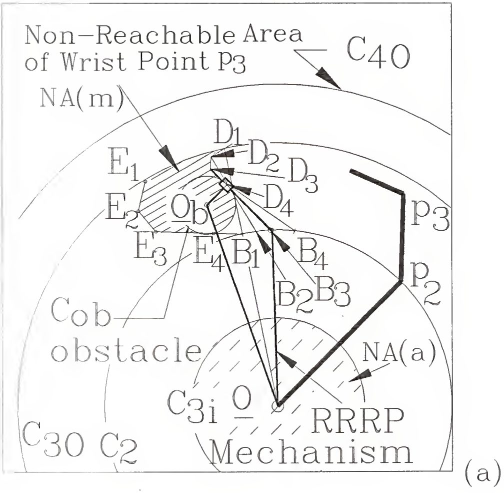


Figure 2.2(a) Determination of the *NA*(a) and *NA*(m) (case 1).

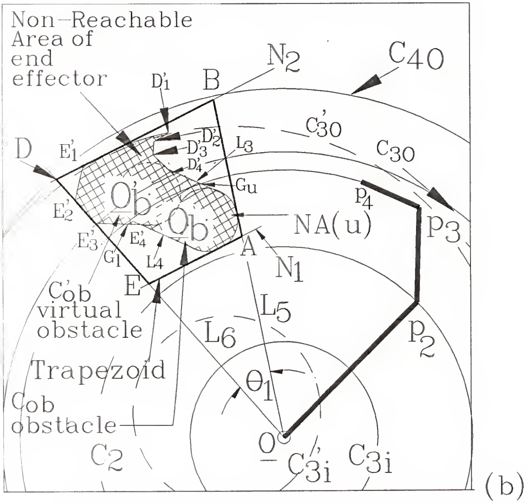


Figure 2.2(b) Determination of the *NA*(u) and the trapezoid T_{ABDE} (case 2).

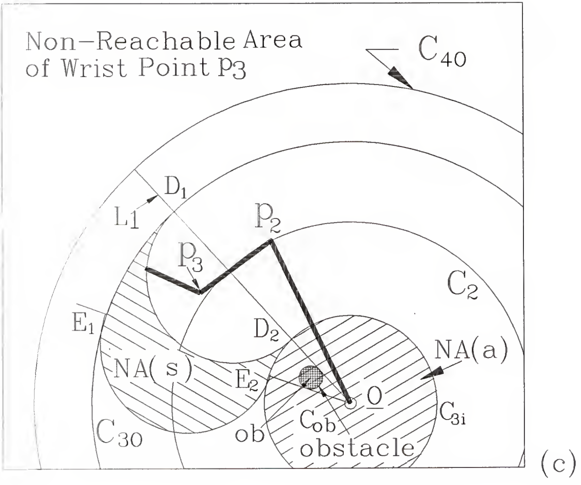


Figure 2.2(c) Determination of the *NA*(a) and *NA*(s) (case 2).

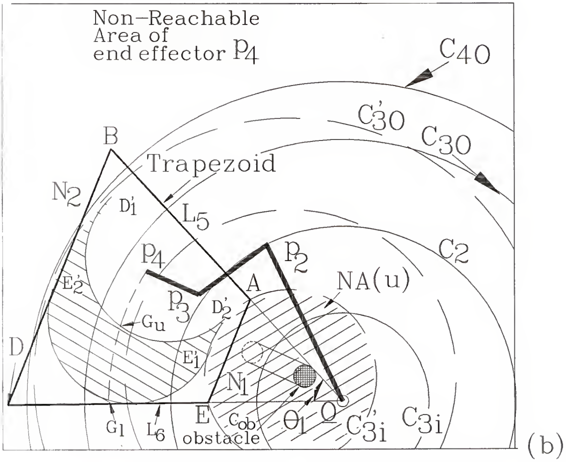


Figure 2.2(d) Determination of the *NA*(u) and the trapezoid T_{ABDE} (case 2).

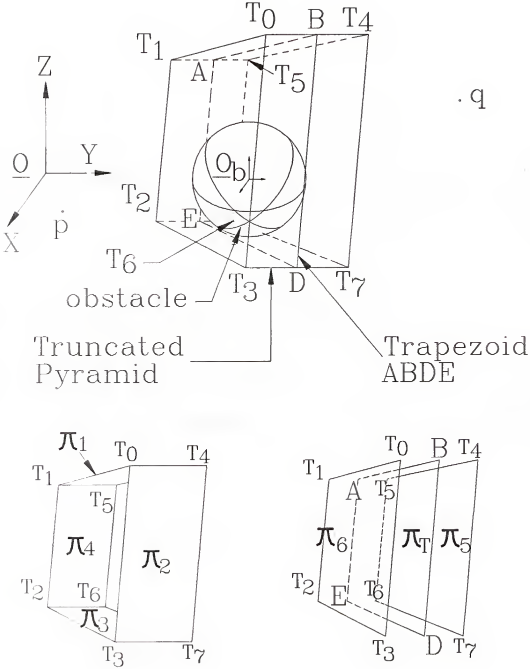


Figure 2.3 Determination of a truncated pyramid.

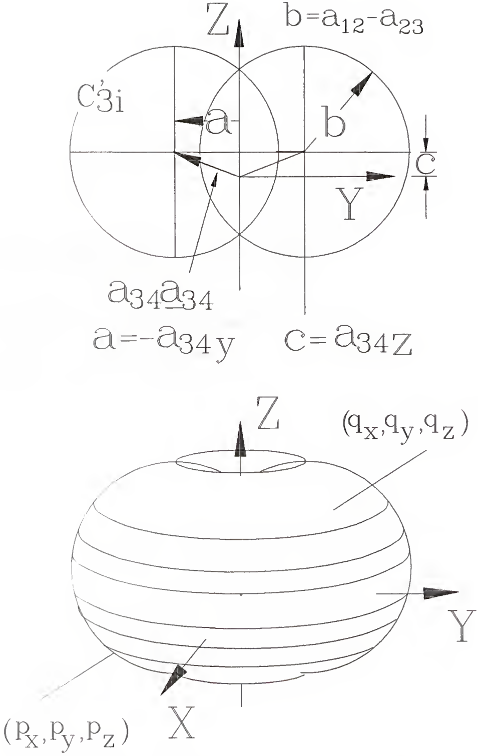


Figure 2.4 Determination of a torus.

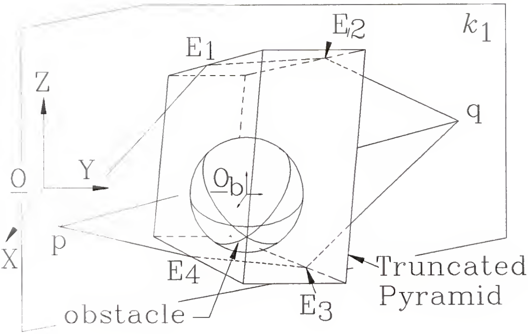
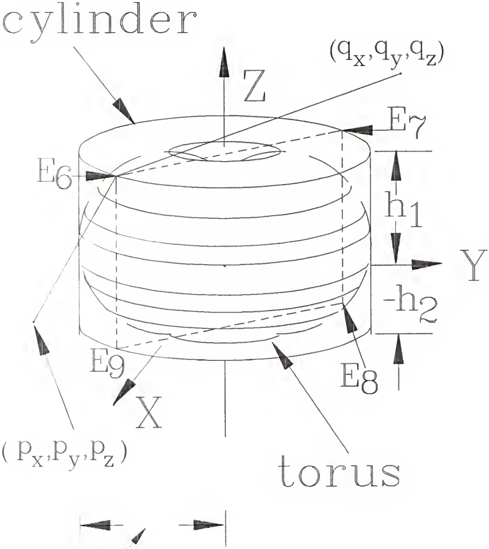


Figure 2.5 Determination of collision free paths for avoiding a truncated pyramid.



$$a_{34y}+(a_{12}-a_{23})$$

$$h_1=a_{34z}+(a_{12}-a_{23})$$

$$h_2=a_{34z}-(a_{12}-a_{23})$$

Figure 2.6 Determination of collision free path for avoiding a torus.

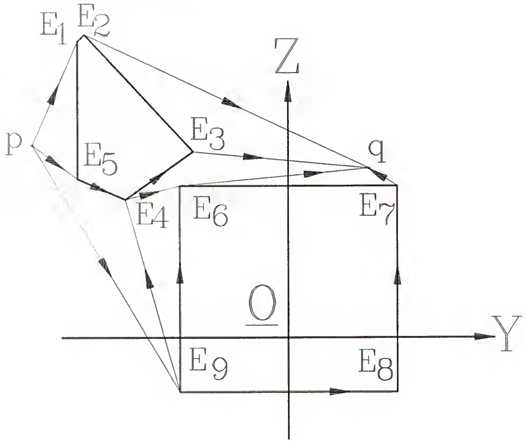


Figure 2.7 Determination of collision free paths for avoiding a truncated pyramid and a torus.

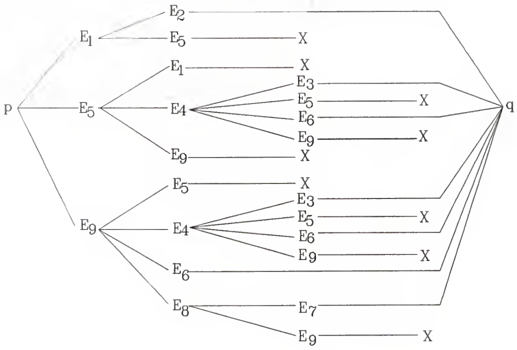


Figure 2.8 The process of determining collision free paths for avoiding a truncated pyramid and a torus.

CHAPTER 3
AUTONOMOUS PATH PLANNING FOR A SPATIAL 4R MANIPULATOR WITH MULTIPLE SPHERICAL OBSTACLES INSIDE THE WORKSPACE

3.1 Introduction

The path planning algorithm for the case of a single spherical obstacle is extended to the case with multiple spherical obstacles inside the workspace. In the previous chapter, the geometry of the non-reachable workspace of the end effector tip of the robot was intensively studied. A truncated pyramid was used to bound the non-reachable workspace of the end effector tip with respect to a single spherical obstacle such that the problem of guiding the robot to avoid a spherical obstacle was reduced to that of moving a point (the end effector tip) while avoiding a truncated pyramid and/or a right circular torus.

In this chapter, the truncated pyramid is transformed into a rectangle in a special coordinate system which is called the *Inclination Angle Coordinate System* (*IACS*). The problem of guiding the robot to avoid spherical obstacles is further reduced to that of moving a point and at the same time avoiding rectangles in the *IACS*. In this way, the complexity of the path planning is reduced from the 3D case to the 2D case in the *IACS*, and the speed of generating the collision free paths is improved significantly. Finally, the free paths in the *IACS* are transformed into the 3D collision free paths in the Cartesian coordinate system.

There are two methods for determining the collision free paths. The computation time for generating 10 collision free paths with 7 spherical obstacles inside the workspace are about 1.5 (method I) and 1.0 seconds (method II) respectively in a Silicon Graphics 4D-70GT workstation. The computation time, 1.5 (method I) and 1.0 seconds (method II), includes computer graphics preparation and generation of 10 collision free paths.

3.2 Determination of Collision Free Paths with Multiple Spherical Obstacles inside the Workspace (Method I)

3.2.1 Transformation of a Truncated Pyramid into a Rectangle in the *IACS*

The truncated pyramid (see Fig. 2.3) is determined by six planes, which are the top (π_1), bottom (π_3), right (π_6), left (π_5), front (π_4) and rear (π_2) faces. These six planes are determined by a trapezoid T_{ABDE} constructed in the vertical plane π_T passing through the center of the spherical obstacle and the robot center $\underline{O}$, which is defined as the point at the intersection of the axes of the first and second joints (see Fig. 2.1). Details of this construction have been given in the previous chapter and will not be repeated here. Briefly, the trapezoid T_{ABDE} is used to enclose the non-reachable area of the end effector tip on the vertical plane π_T. The planes π_1, π_2, π_3, π_4 are drawn passing through the edges of the trapezoid and are perpendicular to the vertical plane π_T. Two vertical planes π_5 and π_6 are drawn through the robot center and tangent to the sides the spherical obstacle. These six planes determine a bounding box, called the truncated pyramid.

The horizontal inclination angles $\beta_{\pi 5}$, $\beta_{\pi 6}$ and $\beta_{\pi T}$ are the deviation angles measured from the vertical plane π_i (which contains the initial configuration of the manipulator) to the vertical planes π_5, π_6 and π_T respectively (see Fig. 3.1). The angles $\beta_{\pi 5}$, $\beta_{\pi 6}$ and $\beta_{\pi T}$ are measured relative to the vertical plane π_i. Further, on the vertical plane π_T, the vertical inclination angles $\alpha_{\pi Tu}$ and $\alpha_{\pi Tl}$ are the deviation angles measured from the horizon to the edges AB and DE of the trapezoid T_{ABDE} respectively. Similarly, on the vertical plane π_5, the inclination angles $\alpha_{\pi 5u}$ and $\alpha_{\pi 5l}$ are the deviation angles measured from the horizon to the edges T_4T_5 and T_6T_7, which are the intersections of the planes π_1 and π_5, π_3 and π_5. On the vertical plane π_6, the inclination angle $\alpha_{\pi 6u}$ and $\alpha_{\pi 6l}$ are the angles measured from the horizon to the edges T_0T_1 and T_2T_3, which are the intersections of the planes π_1 and π_6, π_3 and π_6. However, since the truncated pyramid is symmetrical with respect to the vertical plane π_T, the angle $\alpha_{\pi 5u}$ is equal to $\alpha_{\pi 6u}$, and $\alpha_{\pi 5l}$ is equal to $\alpha_{\pi 6l}$. Further, the larger of the angles

$\alpha_{\pi Tu}$ and $\alpha_{\pi 5u}$ is chosen as the α_U and the smaller of the angles $\alpha_{\pi Tl}$ and $\alpha_{\pi 5l}$ is chosen as the α_L. Here, α_U and α_L (see Fig. 3.2) are the vertical inclination angles, which represent the upper and lower bounds on the Y axis for the rectangle in the *IACS*. Further, $\beta_{\pi 5}$ and $\beta_{\pi 6}$ represent the horizontal inclination angles, which represent the upper and lower bounds on the X axis. In this way, the truncated pyramid can be transformed into a rectangle in the *IACS* with vertices, for example, $V_0(\beta_{\pi 6}, \alpha_U)$, $V_1(\beta_{\pi 6}, \alpha_L)$, $V_2(\beta_{\pi 5}, \alpha_L)$ and $V_3(\beta_{\pi 5}, \alpha_U)$ respectively as shown in Fig. 3.3.

3.2.2 Determination of Free Paths in the *IACS*

In this section, the collision free paths with multiple rectangles in the *IACS* are determined. The spherical obstacles which lie inside the swept volume bounded by the vertical planes π_i, π_f (which contains the final configuration of the manipulator) and the outer boundary of the end effector are considered. The spherical obstacles which lie outside the swept volume are excluded from consideration.

If, for instance, there are seven spherical obstacles lying inside the workspace and the swept volume (see Fig. 3.4), seven corresponding truncated pyramids are determined which contain the seven non-reachable workspaces of the end effector tip with respect to the seven spherical obstacles. Using the algorithm developed above, the seven truncated pyramids are transformed into seven different rectangles in the *IACS* (see Fig. 3.5). The seven rectangles are labeled from 0 through 6.

Further, the initial position p is represented by $(0, \alpha_i; l_i)$, and the final position q is represented by $(\beta_f, \alpha_f; l_f)$, where β_f is the horizontal inclination angle of plane π_f measured relative to π_i, so that $\beta_i = 0$. The angles α_i and α_f are the vertical inclination angles measured on the vertical plane π_i and the vertical plane π_f from the horizon to the initial and final positions. The lengths l_i and l_f are the distance measured from the robot center to the initial and final positions respectively.

After the determination of the rectangles in the *IACS*, free paths are determined from the initial position $p(0, \alpha_i)$ to the target position $q(\beta_f, \alpha_f)$. The determination of the distance l for each point in the path will be discussed in section 3.2.3. The determination of free paths in the *IACS* can be constructed by drawing a line segment from p to q. If the line segment pq does not intersect any edges of the rectangles, the line segment is chosen as the free path in the *IACS*.

However, if the line segment pq intersects any edges of the rectangles, a path searching algorithm is implemented to determine the free paths from p to q in the *IACS*. This can be done by drawing line segments from the initial position p to each vertex V_{ij} of a rectangle i, where i=0,6 and j=0,3. The line segments (for instance, pV_{02}, pV_{03}, pV_{22} and pV_{52}) which do not intersect any of the edges of the rectangles are chosen as the first set of segments of the free paths from the initial position p (see Fig. 3.5). Any line segment which intersects the edges of a rectangle is rejected. The vertices V_{02}, V_{03}, V_{22} and V_{52}, which can be reached by the line segments from the initial position p in the *IACS*, are called the reachable nodes with respect to p. In the same way, the reachable nodes with respect to the final target position q are determined, which are V_{40}, V_{41}, V_{51}, V_{60}, V_{61} and V_{62} respectively.

Following this, from each of the reachable nodes V_{02}, V_{03}, V_{22} and V_{52}, the above procedure is repeated to determine a second set of segments of the free paths, for example, the line segments $V_{03}V_{00}$, $V_{22}V_{21}$, $V_{22}V_{32}$, $V_{22}V_{50}$, $V_{22}V_{53}$, $V_{52}V_{51}$ and $V_{52}V_{53}$. Simultaneously, the vertices V_{00}, V_{21}, V_{32}, V_{50}, V_{51} and V_{53} are compared with the reachable nodes drawn from the target position q, and if any nodes are coincident, then the associated line segments constitute a free path from p to q. If the vertices are not identical to the reachable nodes with respect to the target position q, the procedure is repeated to determine a third set of segments of the free paths. For instance, the vertex V_{51} (see Fig. 3.5) is the reachable node with respect to both the vertex V_{52} and the final position q, and also the vertex V_{52} is the reachable node with respect to the initial position p. Therefore, a free path can be determined from the initial position p through the reachable nodes V_{52} and V_{51}, then to the final position q. The free path is $pV_{52}V_{51}q$.

Further, in order to avoid a node appearing in the same path more than once and to find the shortest path between two nodes, several constraints must be imposed as follows.

(Constraint 1) For the free path such as p ... V_{ij} ... V_{mn} ... q (where i and m are the counters of the number of the rectangles in the *IACS*, and j and n are the counters of the number of the vertices of each rectangle), the absolute value of the X coordinate of a node in the path must be greater than or equal to those of the previous nodes in the same path, that is

$$0 <= \ldots <= |V_{ijx}| \ldots <= |V_{mnx}| <= \ldots <= |\beta_f| . \qquad (3.1)$$

This constraint ensures that the path searching direction is always kept toward the target position q.

Further, any two nodes in the same free path cannot be identical to each other. This avoids a node being repeated in a free path.

(Constraint 2) For a free path, if a node V_{mn} can be reached by a node V_{ij} in one step (for instance, the free path p ...

$V_{ij}V_{mn}$ ... q), it should not be accessed from the node V_{ij} using other routes. This ensures that the path between the nodes is the shortest.

The same procedure with the same constraints is repeated to determine other sets of collision free segments until the final position q is reached. Finally, the free paths from p to q in the *IACS* can be determined by connecting the nodes in the same free path together from the position p to the final position q. Using the above constraints, the number of the possible free paths can be reduced. For instance, in the example, the number of paths was reduced from about 1000 to 146. Following this, 10 free paths which have reasonably short trajectories are displayed in the *IACS*. A trajectory is the sum of line segments connecting the nodes. This offers the operator a selection and he may wish to select the shortest trajectory.

3.2.3 Determination of the Inner and Outer Boundaries of the End Effector Tip

In the previous section, the vertical (α) and horizontal (β) inclination angles of each point in the free paths were determined in the *Inclination Angle Coordinate System* (*IACS*). Here, the upper and lower limits of the distance l from the robot center O to the point with a vertical inclination α are determined. The upper and lower limits will be used to check if the distance l of each point in the free paths lies within the limits. For a different vertical inclination angle α and orientation of the end effector, the upper and lower limits of the end effector are different. Therefore, it is necessary to derive for each point in a free path the upper and lower limits at the vertical inclination angle α.

The upper and lower limits of the end effector tip (the orientation of the end effector is $\underline{a}_{34}$) for a particular vertical inclination angle α can be determined by drawing a line with the vertical inclination angle α passing through the robot center $\underline{O}$ and intersecting the inner circle C_{3i}' and outer circle C_{3O}' at points E_1 and E_2 (see Fig. 3.6). The circles C_{3i}' and C_{3O}' are the translated images of the circles C_{3i} and C_{3O}. The circles C_{3i} and C_{3O} are the inner and outer boundaries of the wrist point of the manipulator. Since it is required that the orientation of the end effector is kept constant during the entire motion, the inner and outer boundaries of the end effector tip are the translation of the inner and outer boundaries of the wrist point with a distance a_{34} in the direction $\underline{a}_{34}$. The distances OE_1 and OE_2 represent the lower and upper limits of the end effector tip respectively in the vertical inclination angle α.

3.2.4 Determination of Collision Free Paths in the Cartesian Coordinate System

Since the free paths from the initial position p(0, α_i; l_i) to the target position q(β_f, α_f; l_f) in the *IACS* are determined, each point (β, α) in the free paths with a suitable distance l from the robot center O will generate subsequently a collision free configuration of the manipulator. The suitable distance l means the distance l must lie between the upper and lower limits of the end effector tip in the vertical inclination angle α. The coordinate of the point in the Cartesian coordinate system can then be determined by the three parameters α, β and l (This will be discussed at the end of this section). The configuration of the manipulator with respect to the point can be generated and will be assured to be collision free with all the spherical obstacles.

The collision free paths in the Cartesian coordinate system can be obtained as follows. For example, the vertices V_{52} and V_{51} of the free path $pV_{52}V_{51}q$ in the *IACS* are expressed as (β_{52}, α_{52}) and (β_{51}, α_{51}) respectively. A collision free path can be determined by firstly changing the vertical inclination angle from α_i to α_{52}, and then the horizontal inclination angle from 0 to β_{52} accordingly. The difference (α_{52}-α_i) and the angle β_{52} are compared, and the larger absolute value is chosen as a reference. It is then divided into n_1 steps with s_1 degree for each step. For example, if $|\alpha_{52}-\alpha_i|>|\beta_{52}|$, then the number of steps n_1 is determined by

$$n_1 = (\text{int})|\alpha_{52} - \alpha_i| + 1, \tag{3.2}$$

and

$$s_1 = (\alpha_{52} - \alpha_i) / n_1. \tag{3.3}$$

The angle β_{52} is also divided into n_1 steps with s_2 degree for each step, where

$$s_2 = \beta_{52} / n_1. \tag{3.4}$$

The same procedure is repeated from the vertex V_{52} to the vertex V_{51} (n_2 steps) and then from the vertex V_{51} to the target position q (n_3 steps).

Further, the distance l which is measured from the robot center $\underline{O}$ to the end effector tip is determined by changing the distance l equally from the initial distance l_i to the final distance l_f with (n_1+n_2+n_3) steps. The distance l at each point is checked to see if it lies within the upper and lower limits with respect to the same vertical inclination angle α. If the distance l lies within the upper and lower limits, the distance l is chosen as the distance of the point. However, if the distance l is greater than the upper limit or smaller than the lower limit, the upper or the lower limit is chosen as the distance l of the point in the path. Figure 3.7 shows the distance l together with upper and lower limits at each instance with respect to the horizontal inclination angle β. Finally, each point represented by $(\beta,\alpha;l)$ is transformed into (x,y,z) in the Cartesian coordinate system (see Fig. 3.8), where

$x = l \cos\alpha \cos\beta^*$, (where $\beta^* = \beta^I - \beta$, and β^I is the horizontal inclination angle of the plane π_i measured relative to the X axis)

$$y = l \cos\alpha \sin\beta^*, \tag{3.5}$$

$z = l \sin\alpha$.

3.2.5 Simulations for the Path Planning Algorithm (Method I)

The algorithm developed above has been successfully implemented on the Silicon Graphics 4D/70GT workstation. Figure 3.9 shows the initial and the final configurations of the spatial 4R manipulator and seven arbitrary spherical obstacles inside the workspace. It is required that a reference point p_4 on the end effector of a spatial 4R manipulator with links $a_{01}=0$, $a_{12}=30$, $a_{23}=15$, $a_{34}=10$ units move from the initial position p(26.2, 32.0, 21.4) to the target position q(23.6, -31.0, -13.0). The locations of the spherical obstacles inside the workspace together with their dimensions are given as follows:

obstacle's center	radius
(28.0, 17.2, 21.8)	4.6
(29.8, 7.6, 21.4)	3.0
(22.6, 5.6, 11.2)	4.0
(25.2, -4.6, 7.4)	3.7
(35.8,-18.0,-12.6)	4.8
(20.4, 1.2,-19.6)	2.8
(34.0,-19.6, 22.0)	5.4

In Fig. 3.9, one of the collision free paths is drawn. The computation time for generating this path with seven spherical obstacles inside the workspace is about 1.5 seconds.

3.3 Determination of Collision Free Paths with Multiple Spherical Obstacles inside the Workspace (Method II)

In this section, the spherical obstacles are divided into two groups, I and II. Two different path planning strategies for the two different groups of obstacles are used together to determine the collision free paths for the end effector tip of the manipulator. This improves the computation speed for generating the collision free paths for the case with multiple spherical obstacles inside the workspace.

3.3.1 Determination of Group I and Group II Spherical Obstacles

The spherical obstacles inside the workspace are divided into two groups according to the locations and radii of the obstacles. The spherical obstacles which lie outside the boundary of joint 2 (the intersection of the link a_{12} and link a_{23}) and lie totally or partially inside the outer boundary of the end effector tip are classified as the group I obstacles. The spherical obstacles which lie inside or intersect the boundary of joint 2 are classified as the group II obstacles.

For a group I obstacle, there are three possible ways to determine the collision free paths of the end effector tip. This can be done by firstly determining a truncated pyramid (see Fig. 2.3), which contains the entire non-reachable workspace of the end effector tip with respect to a spherical obstacle. Then, the collision free paths of the end effector tip can be determined passing through the top (π_1), bottom (π_3) or front (π_4) faces of the truncated pyramid.

For a group II obstacle, there are two possible ways to determine the collision free paths of the end effector tip. The collision free path can be determined either passing through the top (π_1) or bottom (π_3) faces of the truncated pyramid because the front face (π_4) of the truncated pyramid lies inside the inner boundary of the end effector tip. The path which passes through the front face will not be considered.

3.3.2 Path Planning for Multiple Spherical Obstacles (Method II)

Two different path planning strategies for the two different groups of obstacles are used together to determine the collision free paths for the end effector tip of the manipulator.

For instance, there are seven spherical obstacles (see Fig. 3.4) inside the swept volume bounded by the vertical planes π_i (which contains the initial configuration of the manipulator) and π_f (which contains the final configuration of the manipulator), and the outer boundary of the end effector tip. The spherical obstacles which lie outside the swept volume are excluded from consideration. Further, the seven spherical obstacles are divided into two groups using the above method. In this example, the spherical obstacles 0, 1, 4 and 6, which lie outside the boundary of joint 2, are classified as group I obstacles. The spherical obstacles 2, 3 and 5, which lie totally or partially inside the boundary of joint 2, are classified as group II obstacles. The collision free paths are determined by firstly determining the truncated pyramids, each of which contains the non-reachable workspace of the end effector tip with respect to a single spherical obstacle. Then, the truncated pyramids are transformed into rectangles in the *Inclination Angles Coordinate System* (*IACS*) (see section 3.2.1). These rectangles in the *IACS* represent the different spherical obstacles. The rectangles which represent group II obstacles are shaded so as to distinguish them from group I obstacles.

Since the spherical obstacles 2, 3 and 5 represent group II spherical obstacles, free paths which avoid this group are generated from either the tops or the bottoms of the obstacles. This implies that the free paths in the *IACS* should be either passing through the tops or bottoms of the corresponding rectangles. Thus, the rectangles 2, 3 and 5 are said to be the "real" obstacles in the *IACS*.

However, the spherical obstacles 0, 1, 4 and 6, which represent the group I obstacles, can be avoided by withdrawing the end effector of the manipulator close to the robot center $\underline{O}$. The paths in the *IACS* of this category are represented by line segments passing through the rectangles (as opposed to passing over the tops or under the bottoms of the rectangles, as in the above case). Therefore, the rectangles 1, 2 and 5 are said to be the "virtual" obstacles in the *IACS*.

Next, a line segment is drawn from the initial position $p(0,\alpha_i)$ to the final position $q(\beta_f,\alpha_f)$ in the *IACS*. If the line segment pq does not intersect any edges of the rectangles, the line segment is said to be a free path in the *IACS* and the angles β and α for each point in the line segment can be determined. Further, the upper and lower limits from p to q can be determined as well for each point (β,α) (see section 3.2.3). The distance l for each point (β,α) can then be determined within the upper and lower limits. Finally, the point represented by $(\beta,\alpha;l)$ lying on the free path pq can be transformed into a 3D collision free path in the Cartesian coordinate system using equation 3.5. However, if the line segment pq intersects any of the edges, a path planning algorithm for a 2D case is imposed to determine the free paths to avoid the three "real" obstacles (rectangles 2, 3 and 5) instead of all the 7 obstacles in the *IACS*. This can save significant computation time, and makes the algorithm more time efficient. For example, in Fig. 3.10, paths $pV_{23}V_{20}V_{30}q$ and $pV_{22}V_{32}V_{31}q$ are two of the possible free paths in this case. The path $pV_{23}V_{20}V_{30}q$ is chosen for illustration because the trajectory of the path in the *IACS* is the shortest among those possible free paths.

It is clear that the path $pV_{23}V_{20}V_{30}q$ is collision free for the rectangles 2 and 3 (the real obstacles in the *IACS*); however, it is not for the rectangles 0, 1 and 6 (the virtual obstacles). In this case, the intersections of the path $pV_{23}V_{20}V_{30}q$ and the edges of the rectangles 0, 1 and 6 are determined. For example, the intersections of the path $pV_{23}V_{20}V_{30}q$ and the rectangle 0 are points H_0 and H_1. The intersections of the path and the rectangle 1 are points H_2 and H_3. The intersections of the path and the rectangle 6 are points H_4 and H_5.

The coordinates of the intersection points H_0, H_1, H_2, H_3, H_4 and H_5 in the *IACS* are expressed as $H_0(\beta_{H0},\alpha_{H0})$, $H_1(\beta_{H1},\alpha_{H1})$, $H_2(\beta_{H2},\alpha_{H2})$, $H_3(\beta_{H3},\alpha_{H3})$, $H_4(\beta_{H4},\alpha_{H4})$ and $H_5(\beta_{H5},\alpha_{H5})$, where β and α represent the horizontal and vertical inclination angles of a point in the *IACS*. The horizontal and vertical inclination angles are changed from the initial position $p(0,\alpha_i)$, through the intermediate points H_0, H_1, V_{23}, H_2, H_3, V_{20}, V_{30}, H_4 and H_5, then to the final position $q(\beta_f,\alpha_f)$. The step sizes of the angles can be determined using equations 3.2 through 3.4.

After the horizontal and vertical inclination angles of each point are determined, the distance l from the robot center $\underline{O}$ to the end effector tip should be calculated such that each point represented by $(\beta,\alpha;l)$ can be transformed into a point in the Cartesian coordinate system using equation 3.5. The distance l can be derived by determining the upper and lower limits of the distance l for each point (β,α) in the free path $pV_{23}V_{20}V_{30}q$, which has been discussed in section 3.2.3. Since the free path $pV_{23}V_{20}V_{30}q$ passes through the rectangles 0, 1 and 6, the end effector of the manipulator is withdrawn close to the robot center $\underline{O}$ to avoid the spherical obstacles 0, 1 and 6. Therefore, the upper limits between the intersection points H_0 and H_1, H_2 and H_3, and H_4 and H_5 are smaller than the original upper limits. The new upper limits can be determined by drawing lines from the robot center $\underline{O}$ and perpendicular to the front faces (π_4) of the truncated pyramids which represent the spherical obstacles 0, 1 and 6. The lines intersect the front faces of the truncated pyramids at the points G_0, G_1 and G_6. The distance l from the robot center O to the points G_0, G_1 and G_6 are calculated and labelled as l_0, l_1 and l_6. The distance l_0, l_1 and l_6 represent the new upper limits between those intersection points H_0 and H_1, H_2 and H_3, and H_4 and H_5 (see Fig. 3.10).

The new upper limits l_0, l_1 and l_6 together with the original upper limits form the final upper limits for the free path $pV_{23}V_{20}V_{30}q$ (see Fig. 3.11). Another 2D path search algorithm is imposed to determine a smooth path connecting the points $p'(0,l_i)$ and $q'(\beta_f,l_f)$ shown in Fig. 3.11. β_f is the horizontal inclination angle of the final configuration, and l_i and l_f are the distance from the robot center O to the end effector tip of the initial and the final configurations. Also, the path should lie between the upper and lower limits from p' to q'. The 2D path search process starts from the point p' and extends to the point q'. This can be accomplished by drawing a line segment p'q'. If the line segment p'q' does not intersect the upper and lower limits between angles 0 and β_f, the distance l of the end effector tip for each point (β,α) in the free path $pV_{23}V_{20}V_{30}q$ can be determined by selecting the distance l at the angle β, that is, a point (β,l) in the line segment p'q'.

However, if the line segment p'q' intersects the upper and lower limits between the angles 0 and β_f, the distance l of the end effector tip should be changed to the upper or lower limits at the intersection point. For example, the line segment p'q' intersects the upper limits at the angle β_{H0}, where the upper limit is l_0. The new path is then drawn from p' to $H_0'(\beta_{H0}, l_0)$ instead of being drawn from p' to q'. The same procedure is repeated from H_0' to the point q'. The 2D path search process will continue until the point q' is reached. The new path from p' to q' in this case is determined ($p'H_0'H_3'H_5'q'$) and shown in Fig. 3.11. The distance l for each point in the free path $pV_{23}V_{20}V_{30}q$ is calculated according to the new path $p'H_0'H_3'H_5'q'$.

Now, each point in the free path $pV_{23}V_{20}V_{30}q$ is represented by $(\beta, \alpha; l)$ in the *IACS*. The point is then transformed into a point in the Cartesian coordinate system using equation 3.5.

The final result of the collision free path in the Cartesian coordinate system with 7 spherical obstacles inside the workspace is shown in Fig. 3.12. Since the spherical obstacles are divided into two different groups and two different path planning strategies are used together to determine the collision free paths. The speed of generating collision free paths in a Silicon Graphics 4D-70GT workstation is faster (1.0 second) compared with the previous one (1.5 seconds), see Fig. 3.9, which does not impose the new path planning strategies.

The algorithms are further extended to the path planning for multiple cylindrical obstacles inside the workspace, where the cylindrical obstacle can be modelled by a series of spherical obstacles with their centers lying on the central axis of the cylinder.

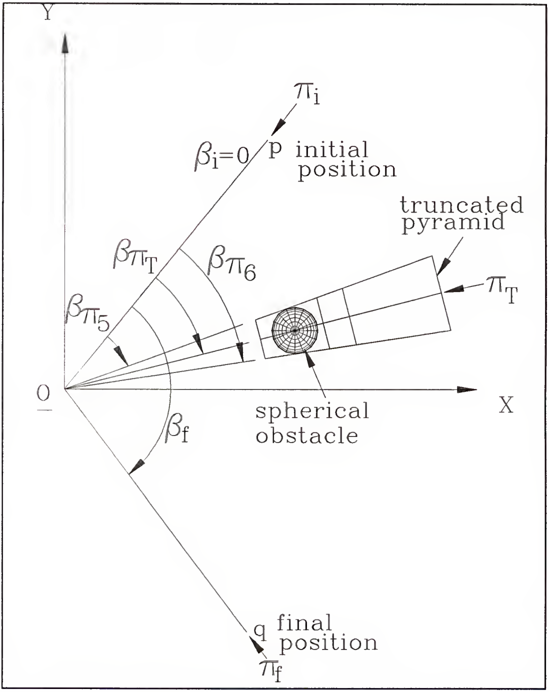


Figure 3.1 Determination of the horizontal inclination angles of the truncated pyramid.

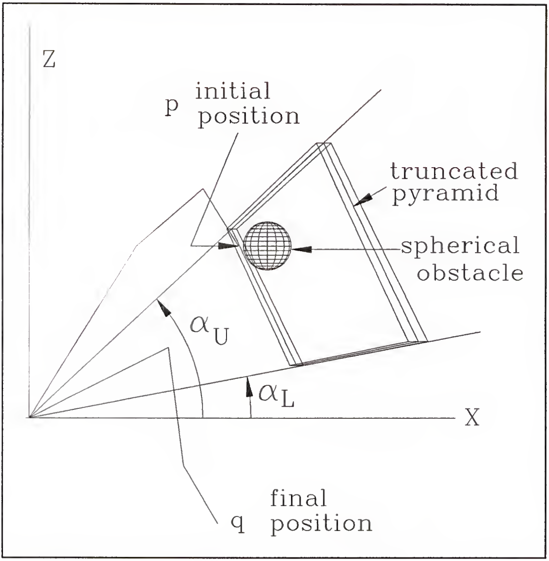


Figure 3.2 Determination of the vertical inclination angles of the truncated pyramid.

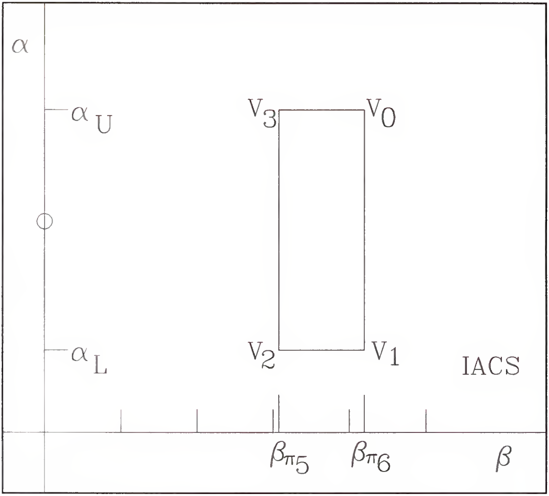


Figure 3.3 Transformation of a truncated pyramid into a rectangle in the Inclination Angle Coordinate system (*IACS*).

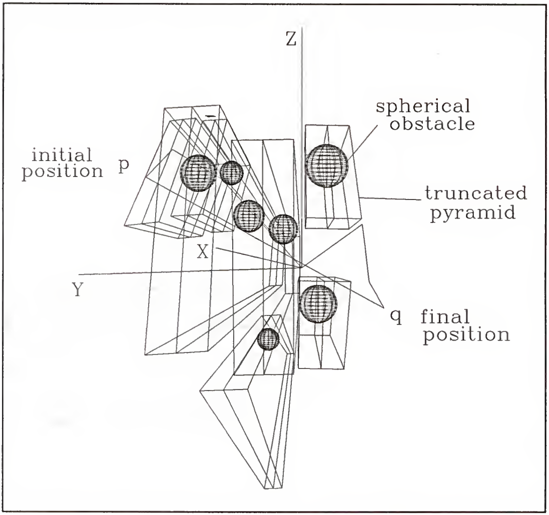


Figure 3.4 Determination of the 7 truncated pyramids with respect to the 7 spherical obstacles.

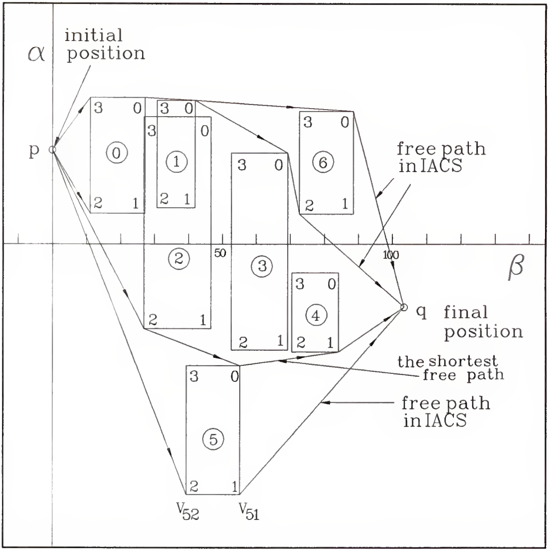


Figure 3.5 Transformation of the 7 truncated pyramids into 7 corresponding rectangles in the *IACS*. Determination of four of the free paths in the *IACS* (method I).

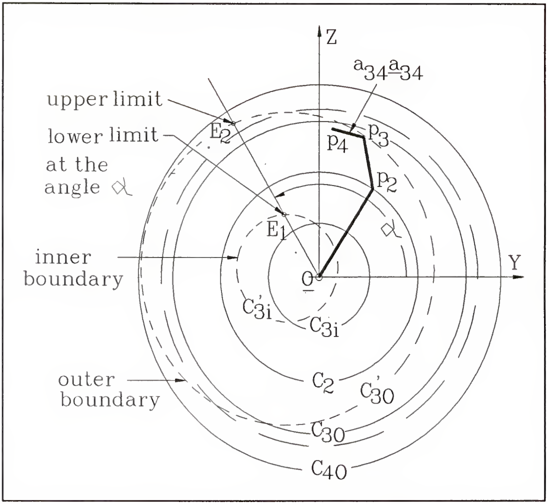


Figure 3.6 Determination of the upper and lower limits of the end effector tip.

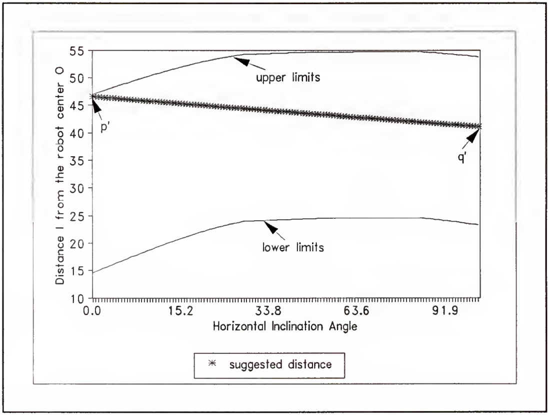


Figure 3.7 Determination of the safe distance *l* for the end effector tip (method I).

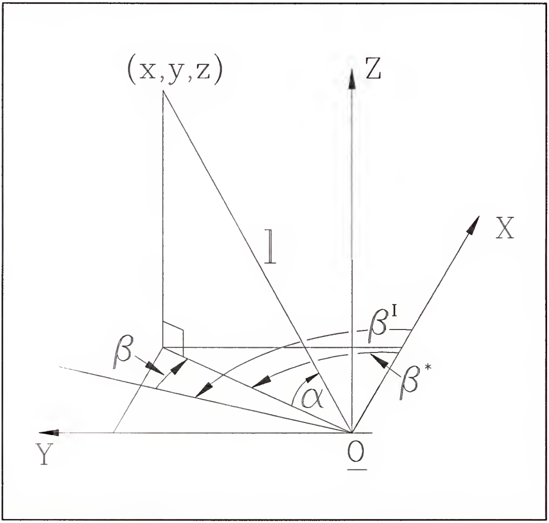


Figure 3.8 Transformation from the polar coordinate system into the Cartesian coordinate system.

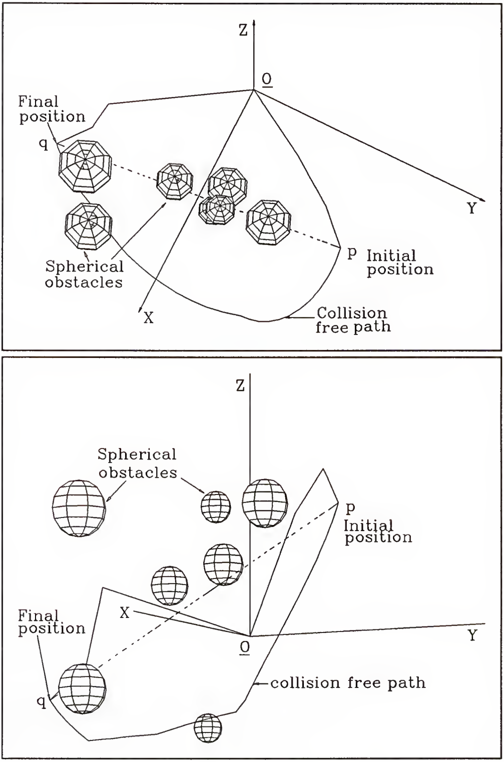


Figure 3.9 Determination of a collision free path in the Cartesian coordinate system (method I).

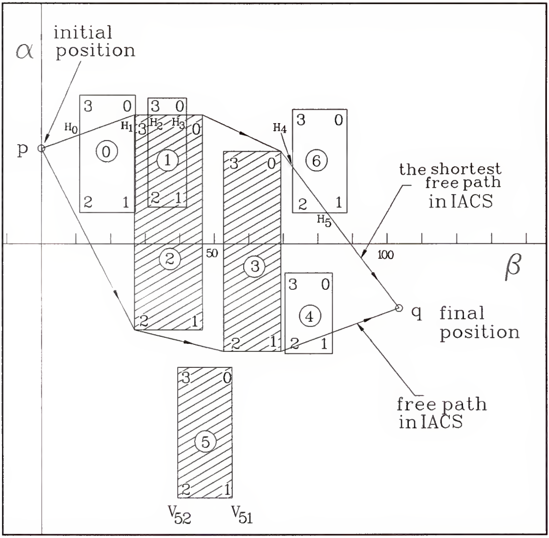


Figure 3.10 Determination of free paths in the IACS with 7 spherical obstacles inside the workspace (method II).

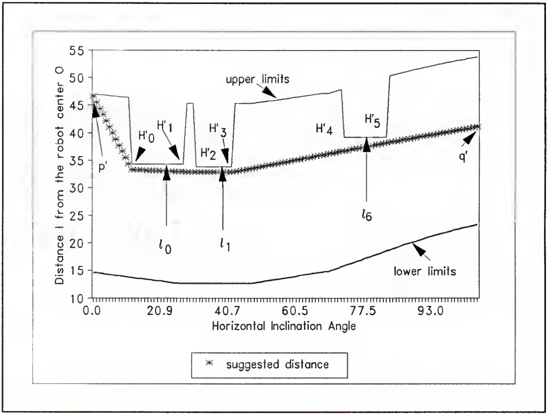


Figure 3.11 Determination of the safe distance l of the end effector tip with 7 spherical obstacles inside the workspace (method II).

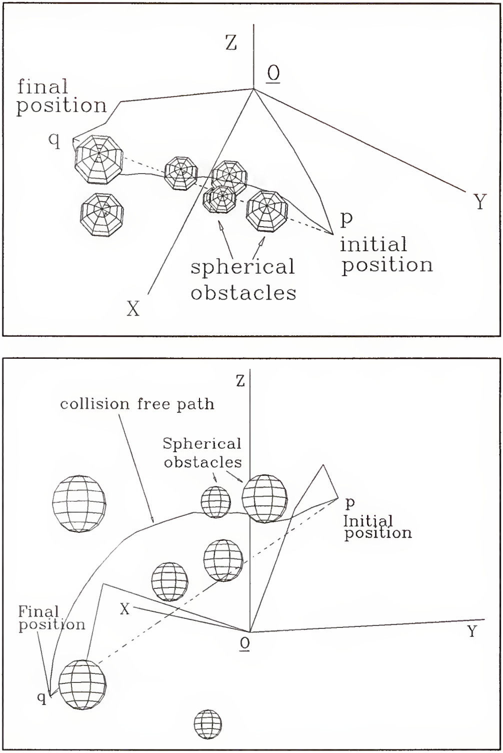


Figure 3.12 Determination of a collision free path of the end effector tip with 7 spherical obstacles inside the workspace (method II).

CHAPTER 4
AUTONOMOUS PATH PLANNING FOR A SPATIAL 4R MANIPULATOR WITH MULTIPLE CYLINDRICAL OBSTACLES INSIDE THE WORKSPACE AND A COMPUTER GRAPHICS SIMULATION ON A $T^3$586 ROBOT

4.1 Introduction

The path planning algorithm for the case of multiple spherical obstacles is extended to the case of multiple cylindrical obstacles inside the workspace. A cylindrical obstacle can be modelled by a series of spherical obstacles with their centers lying on the central axis of the cylinder. Although there is a loss of accuracy by using the spheres to model cylindrical obstacles, the computation speed for the multiple cylindrical obstacles case is as fast as that for the multiple spherical obstacles case. Further, the cylindrical obstacles are not necessarily fixed to ground or ceiling, are neither parallel nor perpendicular to ground. This provides a very practical way for collision free path planning with floating pipes inside the workspace.

Further, the algorithm is modified to guide the spatial $T^3$586 robot around pipes with circular cross section.

4.2 Determination of Collision Free Paths with Multiple Cylindrical Obstacles inside the Workspace

4.2.1 Representation of Cylindrical Obstacles

The cylindrical obstacles, whose lengths and diameters are $L_{cyl,i}$ and $D_{cyl,i}$ respectively, can be modelled as a series of spherical obstacles which contain all the cylinders (see Fig. 4.1). Here, i is the counter of the number of the cylindrical obstacles. The radii ($r_{obs,i}$) and the number ($N_{obs,i}$) of the spherical obstacles which are used to model the ith cylindrical obstacle are determined as follows (see also Fig. 4.1):

$$r_{obs,i} = D_{cyl,i} \;/\; (2^{1/2}), \text{ and} \tag{4.1}$$

$$N_{obs,i} = \text{int}(\; L_{cyl,i} \;/\; D_{cyl,i} \;) + 1. \tag{4.2}$$

The locations of the centers of the spherical obstacles are placed on the central axis of the corresponding cylinder. For example, there are four cylindrical obstacles lying inside the workspace (see Fig. 4.2). The centers of the two end faces of the cylinders are $A_{cyl,i}$ and $B_{cyl,i}$ respectively, where i=0,3. The locations of the centers (obs_x, obs_y, obs_z) of the spherical obstacles for each of the cylinders can be determined as follows:

$$\begin{aligned} obs_{x,j} &= A_{x,cyl,i} + (\; 0.5 + j \;) * D_{cyl,i}, \\ obs_{y,j} &= A_{y,cyl,i} + (\; 0.5 + j \;) * D_{cyl,i}, \text{ and} \\ obs_{x,j} &= A_{z,cyl,i} + (\; 0.5 + j \;) * D_{cyl,i}, \end{aligned} \tag{4.3}$$

where

$i = 0, 3,$ is the counter of the number of the cylindrical obstacles,

$j = 0, N_{obs,i} - 2,$ is the counter of the number of the spherical obstacles needed for the i^{th} cylinder.

When $j = N_{obs,i} - 1,$

$$obs_{x,j} = B_{x,cyl,i} - 0.5 * D_{cyl,i},$$
$$obs_{y,j} = B_{y,cyl,i} - 0.5 * D_{cyl,i}, \text{ and} \qquad (4.4)$$
$$obs_{x,j} = B_{z,cyl,i} - 0.5 * D_{cyl,i}.$$

If the length ($L_{cyl,i}$) of a cylindrical obstacle is smaller than or equal to the diameter ($D_{cyl,i}$) of the cylinder, then the radius and the center of the spherical obstacle can be determined as follows:

$$r_{obs,i} = 0.5 * (D_{cyl,i}^2 + L_{cyl,i}^2)^{1/2}, \text{ and} \qquad (4.5)$$
$$N_{obs,i} = 1. \qquad (4.6)$$
$$obs_{x,0} = 0.5 * (A_{x,cyl,i} + B_{x,cyl,i}),$$
$$obs_{y,0} = 0.5 * (A_{y,cyl,i} + B_{y,cyl,i}), \text{ and} \qquad (4.7)$$
$$obs_{x,0} = 0.5 * (A_{z,cyl,i} + B_{z,cyl,i}).$$

4.2.2 Path Planning for Multiple Cylindrical Obstacles

After the determination of the spherical obstacles which are used to model all the cylindrical obstacles inside the workspace, the path planning algorithms for multiple spherical obstacles are used to determine the collision free paths for the cylindrical obstacle case. Details of the path finding for multiple spherical obstacles inside the workspace have been given in the sections 3.2 and 3.3 and will not be repeated here. Briefly, this can be done by firstly determining the truncated pyramids, each of them contains the non-reachable workspace of the end effector tip for the corresponding spherical obstacle. The truncated pyramids are, then, transformed into rectangles in the *Inclination Coordinate System (IACS)*. For example, there are four cylindrical obstacles inside the workspace (see Fig. 4.2). The locations and dimensions of the cylindrical obstacles are given as follows:

Length	diameter	end point A	end point B
9.1455	5.0	(33.3, 15.0, 17.9)	(28.3, 10.0, 12.1)
28.723	3.5	(44.8, 20.0, 17.1)	(39.8, 0.0, -2.9)
32.015	6.0	(33.5, -6.4, 10.9)	(28.5, 3.6, 40.9)
36.400	4.5	(58.0,-10.0, 10.2)	(23.0,-10.0, 0.2)

The four cylindrical obstacles are contained by 26 spherical obstacles. Further, the spherical obstacles are represented by 17 rectangles (9 of the spherical obstacles lie outside the workspace) in the *IACS* (see Fig. 4.3). Three of the rectangles are shaded, which are the "real" obstacles in the *IACS* (see section 3.3.2). A 2D path planning algorithm is imposed to determine free paths in the *IACS*. After the determination of the free paths, the path $pV_0V_1V_2q$ with the shortest trajectory is chosen, as illustrated in Fig. 4.3. The horizontal and vertical inclination angles (β,α) for each point can, then, be determined according to the selected free path $pV_0V_1V_2q$ in the *IACS*.

Next, the distance l from the robot center O to the end effector tip is determined, which should lie between the upper and lower limits of the end effector tip for each point (β,α). The upper and lower limits from $\beta=0$ to $\beta=\beta_f$ are determined and shown in Fig. 4.4 (see also section 3.3.2 for detail). Another 2D path search algorithm is imposed to determine a smooth path from $p'(0,l_i)$ to $q'(\beta_f,l_f)$, for example, $p'H_6'H_7'q'$. Then, the distance l for each point (β,α) can be determined according to the suggested path $p'H_6'H_7'q'$, where for each angle β there is a corresponding distance l.

After the determination of the horizontal (β) and vertical (α) inclination angles and the distance l for each angle β, the collision free path from the initial position p to the final position q is determined and the points in the collision free path are represented by the three parameters $(\beta,\alpha;l)$. The point represented by $(\beta,\alpha;l)$ can be transformed into a corresponding (x,y,z) in the Cartesian coordinate system using equation 3.5.

The final result of the collision free path in the Cartesian coordinate system with 4 cylindrical obstacles inside the workspace is shown in Fig. 4.2 (top and side views). The computation time for determining the collision free path is about 1 second.

In case that the initial position p (or final position q) lies inside the truncated pyramid, the reachable workspaces which are ignored because of using the truncated pyramid to contain the non-reachable workspace become very important. When the truncated pyramid is transformed into a rectangle in the *IACS*, the initial position (or final position) is also transformed into a point in the *IACS*. The point must lie inside the rectangle (see Fig. 4.5). In Fig. 4.5, the initial position p and final position q lie inside four "virtual" rectangles drawn by broken lines.

The boundary of the non-reachable workspace of the end effector tip for the corresponding spherical obstacle is calculated instead of being contained by a truncated pyramid. This can be done by drawing a circle with center $\underline{O}$ and radius l_{p4} (the distance from the robot center $\underline{O}$ to the end effector tip $\underline{p}_4$, see Fig. 2.2(b)) on the vertical plane π_T passing through the Z axis and the center of the spherical obstacle. The circle intersects the boundary of the *NA* for the end effector tip at two points G_u and G_l. The horizontal inclination angle $\beta_{\pi T}$ is the deviation angle measured from the vertical plane π_i (which contains the initial configuration of the manipulator) to the vertical plane π_T (see Fig. 3.1). The vertical inclination angles α_{Gu} and α_{Gl} are the deviation angles measured from the horizon to the points G_u and G_l on the vertical plane. Then, $(\beta_{\pi T}, \alpha_{Gu})$ and $(\beta_{\pi T}, \alpha_{Gl})$ represent two points in the *IACS*.

In the same way, six more vertical planes are drawn passing through the Z axis and intersecting the spherical obstacle. Three of the six vertical planes lie equally between the vertical planes π_5 and π_T of the truncated pyramid, and three other vertical planes lie equally between the π_6 and π_T. Two different points are determined on each of the six vertical planes and each of the vertical planes π_5, π_6 and π_T. A conic can be drawn by connecting these 18 points in the *IACS* (see Fig. 4.5). The radius l_{p4} (the distance from the robot center $\underline{O}$ to the end effector tip $\underline{p}_4$, see Fig. 2.2(b)) is used to determine all the 18 points of the conic, and hence the distance l for each point on the collision free path between the range $\beta_{\pi 5}$ and $\beta_{\pi 6}$ is fixed to be l_{p4}.

The same procedure is repeated if the initial position p (or final position q) lies inside the other truncated pyramids. Figure 4.5 shows that four conics are generated for the initial position p and final position q respectively. After the determination of these conics and rectangles, the collision free paths for the end effector tip can be determined using the algorithm shown in section 3.3.2. Figure 4.5 shows a free path from initial position p to the final position q in the *IACS*. The safe distance l for each point (β,α) on the free path from p to q can be determined from figure 4.6, where l_p represent the distance from the robot center O to the end effector tip for the initial configuration and l_q for the final configuration. The final result of the collision free path in the Cartesian coordinate system for this special case with 4 cylindrical obstacles inside the workspace is shown in Fig. 4.7 (top and side views).

4.3 Computer Graphics Simulation on $T^3$586 Robot

The algorithms are modified to guide the spatial $T^3$586 robot around pipes with circular cross section. Since the spatial $T^3$586 robot has real dimensions for each links, it is assumed that the robot's links are bounded by cylinders with lengths $L_{link,i}$ and diameters $D_{link,i}$. The lengths $L_{link,i}$, where i=0,3, are assumed to be equal to the lengths of the links a_{01}, a_{12}, a_{23}, and a_{34}. The diameters $D_{link,i}$, where i=0,3, are determined according to the dimension of the upper arm of a $T^3$586 robot. The cross section of the upper arm of the $T^3$586 robot is shown in Fig. 4.8. It is assumed that the cross section is bounded by a rectangle with w_L in width and h_L in height. Then, the diameters $D_{link,i}$ of the bounding cylinders are determined by

$$D_{link,i} = (w_L^2 + h_L^2)^{1/2}, \qquad (4.8)$$

where

$$i = 0, 3.$$

The bounding cylinders and the $T^3$586 robot are shown together in Fig. 4.9.

The radii of spherical obstacles are enlarged according to the diameters $D_{link,i}$, that is, the diameter of each spherical obstacle is enlarged by adding $D_{link,i}$. In this way, the robot's links are shrunk into line segments and the algorithm developed above can be used for determining the collision free paths for the spatial $T^3$586 robot. The computation time of 1 second is the same as before for the multiple cylindrical obstacles.

The algorithm has been successfully developed and implemented on the Silicon Graphics 4D-70GT workstation. In order to permit a user to verify the results easily, the computer graphics simulation is designed as an interactive program which enables a user to change arbitrarily the number, locations and sizes of the spherical and cylindrical obstacles, and the initial and target positions of the end effector tip. In the same way, the orientation of the end effector can be specified arbitrarily by using the mouse device. Furthermore, if collision free paths for the end effector tip cannot be found, the user will be asked to change locations and sizes of the spherical and cylindrical obstacles for the next path searching procedure.

Figure 4.10 shows the initial configuration of the spatial $T^3$586 robot with 10 cylindrical obstacles inside the workspace. Figure 4.11 shows the final configuration of this spatial robot. A collision free path with 10 cylindrical obstacles inside the workspace can be determined in about 2 seconds. The final collision free path together with an intermediate configuration of the spatial $T^3$586 robot are shown in Fig. 4.12.

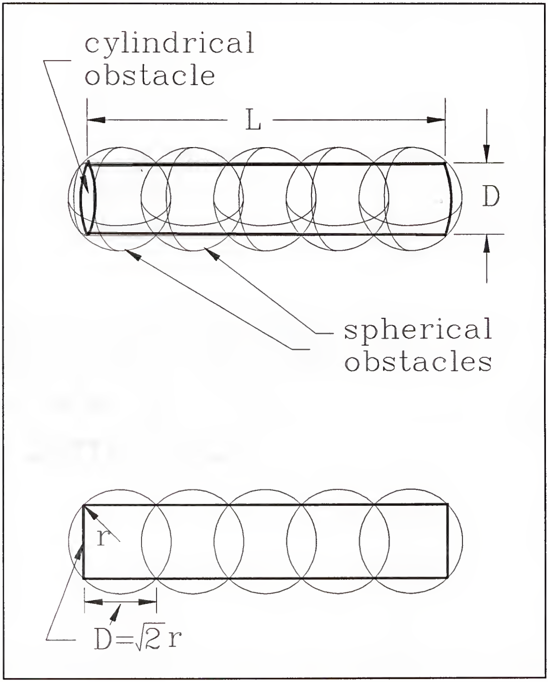


Figure 4.1 A cylindrical obstacle is modeled by a series of spherical obstacles.

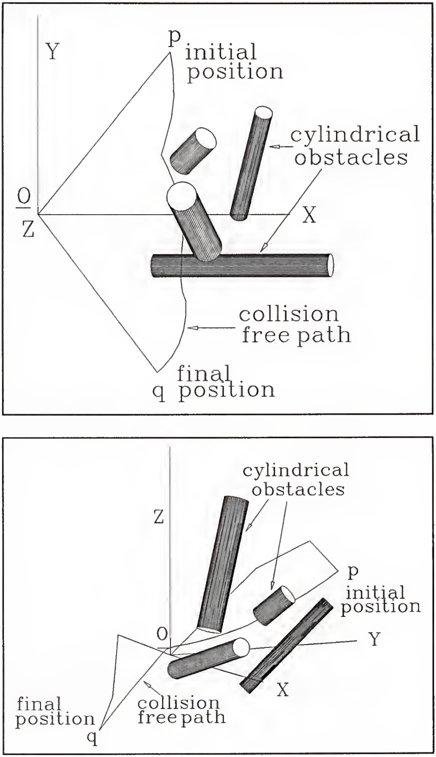


Figure 4.2 Determination of a collision free path of the end effector tip with 4 cylindrical obstacles inside the workspace.

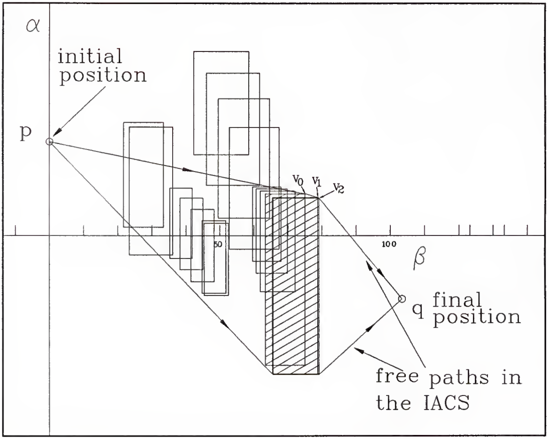


Figure 4.3 Determination of free paths in the IACS with 4 cylindrical obstacles inside the workspace.

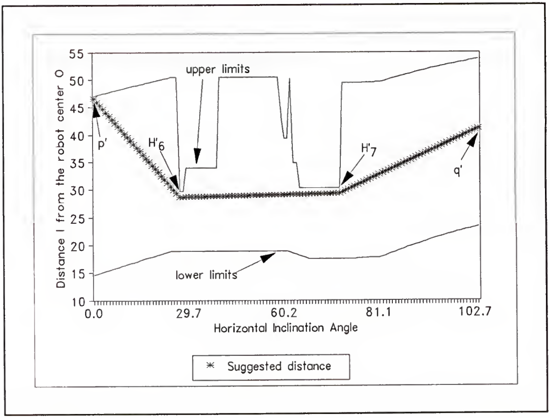


Figure 4.4 Determination of the safe distance *l* of the end effector tip with 4 cylindrical obstacles inside the workspace.

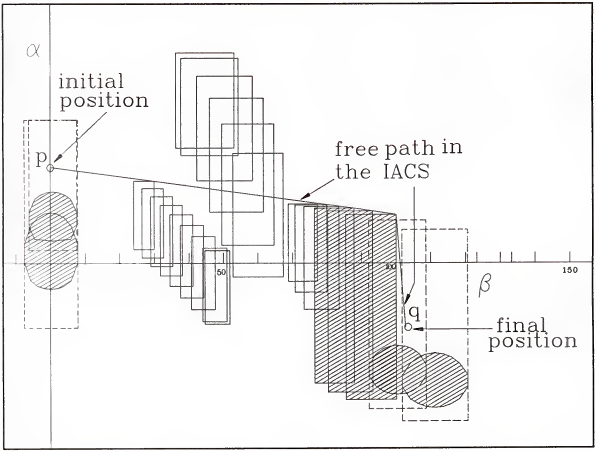


Figure 4.5 Determination of a free path in the *IACS* with 4 cylindrical obstacles inside the workspace.

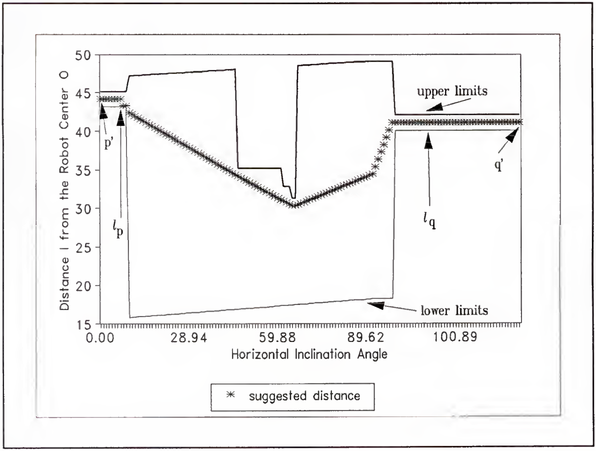


Figure 4.6 Determination of the safe distance *l* of the end effector tip with 4 cylindrical obstacles inside the workspace.

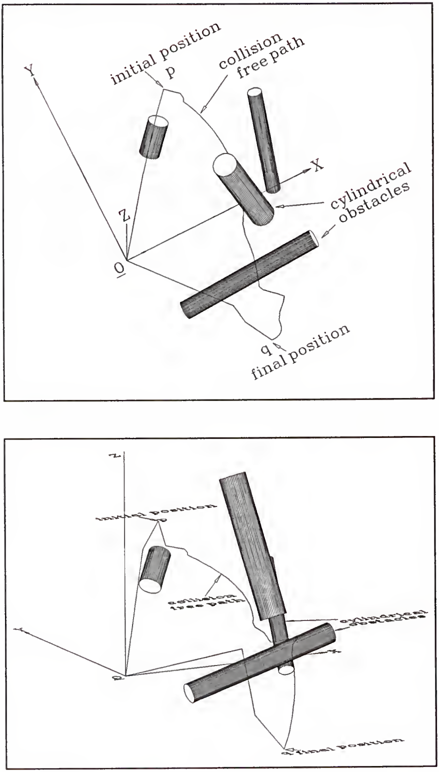


Figure 4.7 Determination of a collision free path for the end effector tip with 4 cylindrical obstacles inside the workspace (top and side views).

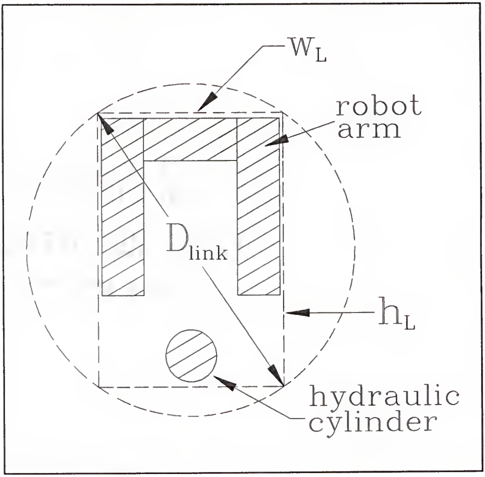


Figure 4.8 A cross section of the upper arm of the spatial $T^3$586 robot.

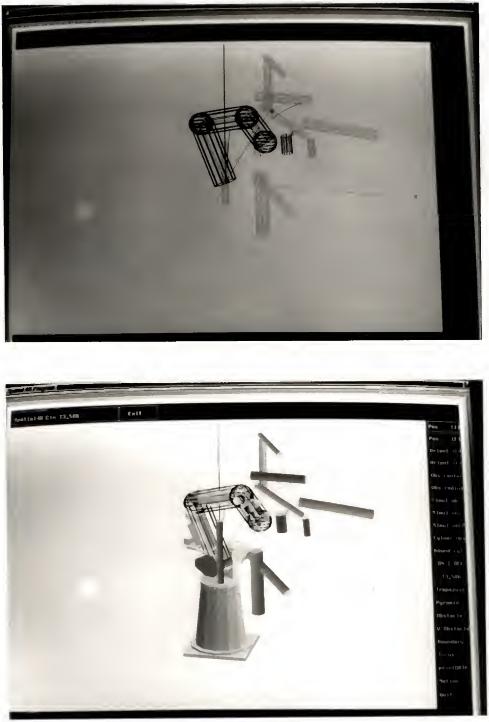
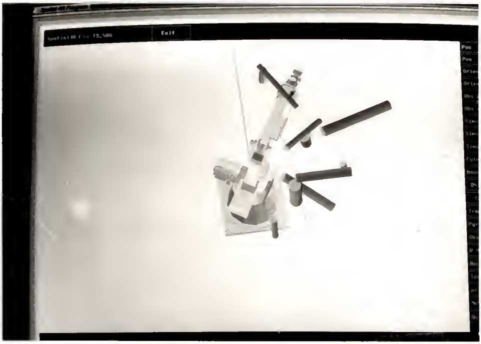


Figure 4.9 The bounding cylinders for the spatial $T^3$586 robot.



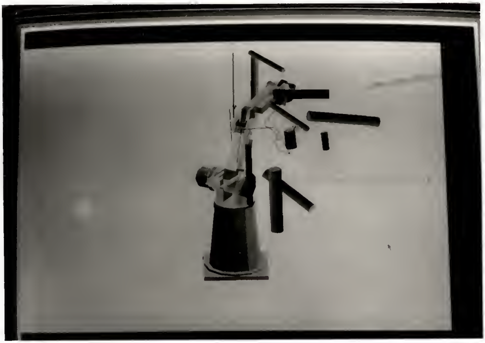
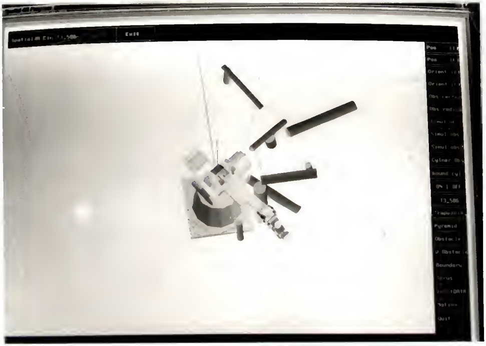


Figure 4.10 The initial configuration of the spatial $T^3$586 robot (top and side views).



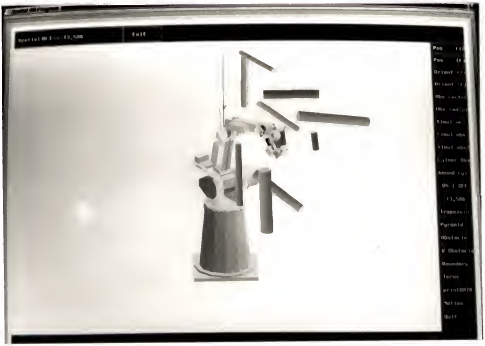


Figure 4.11 The final configuration of the spatial T^3586 robot (top and side views).

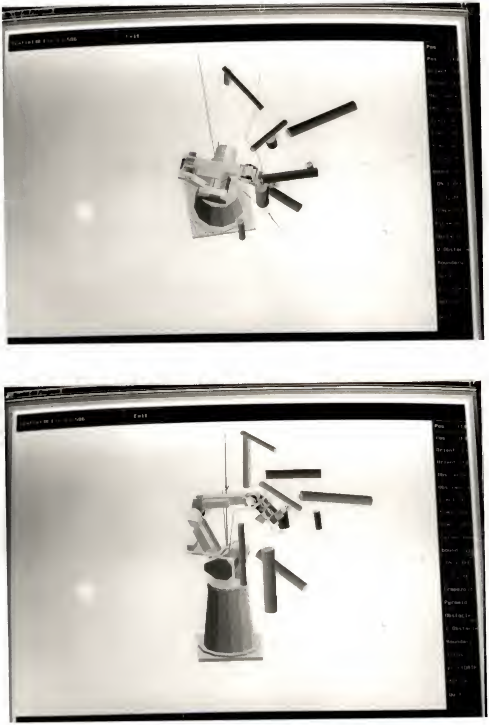


Figure 4.12 The intermediate configuration of the spatial $T^3$586 robot and a collision free path of the end effector tip with 10 cylindrical obstacles inside the workspace (top and side views).

CHAPTER 5
CONCLUSIONS AND FUTURE WORKS

Algorithms which can rapidly generate collision free paths for the end effector tip of a spatial 4R robot (see Fig. 2.1) with multiple spherical obstacles or cylindrical obstacles inside the workspace have been successfully developed.

The non-reachable workspace which cannot be reached by the end effector tip because of the existence of a spherical obstacle is considered to be bounded by a relatively simple truncated pyramid. The other non-reachable workspace of the end effector tip is a central void, which can be represented by a right circular torus.

With the representations of the truncated pyramid and the right circular torus, the problem of guiding a spatial 4R manipulator while avoiding spherical obstacles is reduced to that of moving a point (tip of the end effector) and at the same time avoiding truncated pyramids and/or a right circular torus. Although the problem is reduced to the path planning of a point (the end effector tip), the collisions between the links of the manipulator and the spherical obstacles have already been considered.

Further, the truncated pyramids are transformed into rectangles in the *Inclination Angle Coordinate System* (*IACS*). The path planning problem is further reduced to moving a point and at the same time avoiding rectangles in the *IACS*. The free paths determined in the *IACS* are then transformed into the final 3D collision free paths in the Cartesian coordinate system. In this way, the complexity of the path planning is reduced from the 3D case to the 2D case without losing the characteristics of a 3D path planning. And, the speed of generating the collision free paths is improved significantly.

The algorithms are further extended to the case with multiple cylindrical obstacles inside the workspace, where the cylindrical obstacles are not necessarily fixed to ground or ceiling, and are neither parallel nor perpendicular to ground. This provides a practical way for determining collision free paths for the end effector tip with floating pipes inside the workspace.

The algorithms have been successfully implemented on the Silicon Graphics 4D-70GT workstation. The computation time for generating 10 collision free paths with 7 spherical obstacles or 10 cylindrical obstacles inside the workspace is about 1 or 2 seconds. The computation time (1 or 2 seconds) includes the computer graphics preparation and the generation of 10 collision free paths.

Also, the algorithms are modified to guide a spatial $T^3$586 robot around pipes with circular cross sections. Further, the computer graphics simulation is designed as an interactive program, which enables a user to change information using a mouse device and to verify results easily.

The research may be extended as follows:

1) Changing the orientation of the end effector. In the algorithms, it has been assumed that the end effector of the manipulator moves with constant orientation during the entire motion. In the future, the algorithms may be extended to be able to change the orientation of the end effector while avoiding obstacles.

2) Path planning for moving spherical or cylindrical obstacles. Since the computation speed of the algorithms is fast, it is also expected to extend the path planning algorithms to the case with moving spherical or cylindrical obstacles inside the workspace.

3) Path planning for a pair of cooperating robots moving between spherical or cylindrical obstacles. If (2) can be done successfully, the collision free paths between two or more cooperating robots working in an overlapped region of the workspaces can also be determined. Here, the links of the secondary robots can be modelled by a series of spherical obstacles.

4) Path planning for picking and placing an object. It is important that the robot can perform the tasks, such as

picking and placing an object; however, the boundaries of the non-reachable workspace of the end effector tip will become much more complicated, depending on the shape and size of the object the robot grabs.

5) Modifying the algorithms to guide the spatial PUMA, $T^3$776, and ATMS robots around pipes with circular cross-sections. The algorithms may be modified to guide the spatial PUMA, $T^3$776, and ATMS robots around pipes with circular cross-sections. Here, the non-reachable workspace of the end effector tip could be determined according to the shapes and dimensions of the different types of spatial robots.

6) Implementation of the algorithms using a real robot for on line collision free motion planning. The best way to evaluate the algorithms is to implement the algorithms onto a real robot to perform on line collision free motion planning. However, this should include much knowledge and experience in control theories. It is, in the author's opinion, the most challenging task for the future work.

The author recognizes that the algorithms have some constraints when there exist many obstacles which cannot be modelled easily by spheres. For example, a thin plate, a long wire, or the object with special shape, etc. However, the results presented have much practical application for guiding robot manipulator around pipes with circular cross sections. Such workspaces are encountered in nuclear power plants and various industries.

REFERENCES

[1] Udupa S. M., "Collision Detection And Avoidance In Computer Controlled Manipulators," Fifth International Joint Conference on Artificial Intelligence, MIT, Cambridge, Massachusetts, August 1977, pp. 737-748.

[2] Lozano-Perez T., "Task Planning," Robot Motion, The MIT Press Series in Artificial Intelligence, Cambridge, Massachusetts, 1982, pp. 473-498.

[3] Lozano-Perez T., "Automatic Planning of Manipulator Transfer Movements," Robot Motion, The MIT Press Series in Artificial Intelligence, Cambridge, Massachusetts, 1982, pp. 499-535.

[4] Lozano-Perez T., "Motion Planning for Simple Robot Manipulators," The 3rd International Symposium on Robotics Research, The MIT Press, Cambridge, Massachusetts, 1986, pp. 133-140.

[5] Lozano-Perez T., "A Simple Motion-Planning Algorithm for General Robot Manipulators," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3, June 1987, pp. 224-238.

[6] Khatib O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," IEEE International Journal of Robotics and Automation, March 1985, pp. 500-505.

[7] Lumelsky V.J., "On Non-Heuristic Motion Planning in Unknown Environment," IFAC Symposium on Robot Control (SYROCO'85), Barcelona, Spain, November 1985.

[8] Cheung E. and Lumelsky V., "Motion Planning for Robot Arm Manipulators with Proximity Sensing," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp.740-745.

[9] Lumelsky V. and Sun K., "A Unified Methodology for Motion Planning with Uncertainly for 2D and 3D Two-Link Robot Arm Manipulators," International Journal of Robotics Research, Vol. 9, No. 5, October 1990, pp. 89-104.

[10] Young L. and Duffy J., "A Theory for the Articulation of Planar Robots: Part I--Kinematic Analysis for the Flexure and the Parallel Operation of Robots," ASME Design Engineering Technical Conference, Columbus, Ohio, October 5-8, 1986. 86-DET-75.

[11] Young L. and Duffy J., "A Theory for the Articulation of Planar Robots: Part II--Motion Planning Procedure for Interference Avoidance," ASME Design Engineering Technical Conference, Columbus, Ohio, October 5-8, 1986, 86-DET-76.

[12] Choi Y.J., Crane C.D., Matthew G.K., Duffy J., "The Geometry of Interference with Application to Obstacle Avoidance," The 1990 ASME Design Technical Conference - 21st Biennial Mechanisms Conference, Chicago, IL, DE-Vol. 24, Sept. 16-19, 1990, pp. 327-336.

[13] Tseng C.S, "Rapid Generation of Collision-Free Paths for Robot Manipulator with Computer Graphics Animation," Ph.D. Dissertation, University of Florida, 1987.

[14] Shieh M.D. and Duffy J., "Autonomous Rectilinear Motion Planning for a Spatial Robot--Path Planning for a Spatial 4R Manipulator with a Single Spherical Obstacle Inside the Workspace," The 1990 ASME Design Technical Conference - 21st Biennial Mechanisms Conference, Chicago, IL, DE-Vol. 24, Sept. 16-19, 1990, pp. 297-302.

[15] Shiller Z. and Dubowsky S., " Global Time Optimal Motions of Robotic Manipulators in the Presence of Obstacles," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 370-375.

[16] Shiller Z. and Dubowsky S., "Robot Path Planning with Obstacles, Actuator, Gripper, and Payload Constraints," International Journal of Robotics Research, Vol. 8, No. 6, December 1989, pp. 3-18.

[17] Chen Y.C. and Vidyasagar M., "Optimal Trajectory Planning for Planar-n-link Revolute Manipulators in the Presence of Obstacles," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, 202-208.

[18] Meyer W. and Benedict P., "Path Planning and the Geometry of Joint Space Obstacles," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 215-219.

[19] Khosla P. and Volpe R., "Superquadric Artificial Potentials for Obstacle Avoidance and Approach," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 1778-1784.

[20] Hwang Y.K., Ahuja N., "Path Planning Using A Potential Field Representation," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, 648-649.

[21] Xia K., Jiang S., and Lii L., "Obstacle Avoidance Path Planning of A manipulator," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 646-647.

[22] Muck K.L., "Motion Planning in Constraint Space," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 633-635.

[23] Chen J.L. and Duffy J., "An Analysis for Rectilinear Parallel Operation of a Pair of Spatial Manipulators: Part I--Motion Capability," Trends and Developments in Mechanisms, Machines, and Robotics-1988, Vol. 3, Sept. 1988, pp. 393-400.

[24] Chen J.L. and Duffy J., "An Analysis for Rectilinear Parallel Operation of a Pair of Spatial Manipulators: Part II--Range of End Effector Orientation," Trends and Developments in Mechanisms, Machines, Robotics-1988, Vol. 3, Sept. 1988, pp. 393-400.

[25] Zaharakis S.C. and Guez A., "Time Optimal Navigation VIA Slack Time Sets," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 650-651.

[26] Chang Y.H., Lee T.T., and Liu C.H., "On-Line Cartesian Path Trajectory Planning for Robot Manipulators," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 62-67.

[27] Chien Y.P., Koivo A.J., Lee B.H., "On-line Generation of Collision Free Trajectories for Multiple Robots," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 209-214.

[28] Eltimsahy A.H. and Yang W.S., "Near Minimum-Time Control of Robotic Manipulator with Obstacles in the Workspace," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 358-363.

[29] Kant K. and Zucker S., "Planning Collision-Free Trajectories in Time-Varying Environments: A Two-Level Hierarchy," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 1644-1649.

[30] Fujimura K. and Samet H., "Path Planning Among Moving Obstacles Using Spatial Indexing," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 1662-1667.

[31] Avnaim F., Boissonnat J.. and Faverjon. B., "A Practical Exact Motion Planning Algorithm for Polygonal Objects Amidst Polygonal Obstacles," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 1656-1661.

[32] Rao N.S.V., Iyengar S.S. and deSaussure G., "The VIST problem: visibility Graphic-Based Solution," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 1650-1655.

[33] Kanayama Y., "Least Cost Paths with Algebraic Cost Functions," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 75-80.

[34] Brooks R.A., "Planning Collision Free Motions for Pick and Place Operations," The First International Robotics Research Symposium, The MIT Press, Cambridge, Massachusetts, 1984, pp. 5-38.

[35] Gouzenes L., "Strategies for Solving Collision-free Trajectories Problems for Mobile and Manipulator Robots," International Journal of Robotics Research, Vol. 3, No. 4, Winter 1984, pp. 51-65.

[36] Maciejewski A. and Klein C., "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," International Journal of Robotics Research, Vol. 4, No. 3, Fall 1985.

[37] Myers J.K., "A Robotic Simulator with Collision Detection: RCODE," Ist Annual Workshop on Robotics & Expert System, June 1985, pp. 205-213.

[38] Faverjon B., "Obstacle Avoidance using an Octree in the Configuration Space of a Manipulator," IEEE Computer Society International Conference on Robotics, Atlanta, GA, March 1984, pp. 504-512.

[39] Yamada Y., Tsuchida N. and Ueda M., "Control of an Industrial Robot Manipulator with Four Articulations to Continue Tasks while Avoiding Obstacles," 14th International Symposium on Industrial Robots, 7th International Conference on Industrial Robot Technology, Gothenburg, Sweden, Oct. 2-4, 1984, pp. 139-207.

[40] Angeles J. and Akhras R., "Cartesian Trajectory Planing for 3-DOF spherical Wrists," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp 68-74.

[41] Jun S. and Shin K.G., "A Probablistic Approach to Collision-Free Robot Path Planning," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 220-225.

[42] Kabuka M., Glaskowski P. and Miranda J., "A Flexible High Performance Robot Arm Controller," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 628-629.

[43] Kyriakopoulos K.J. and Saridis G.N., "Minimum Jerk Path Generation," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 364-369.

[44] Egeland O. and Lunde E., "Trajectory Generation for Manipulator Using Linear Quadratic Optimal Tracking," Proceedings 1988 IEEE International Conference on Robotics and Automation, Philadelphia, Pennsylvania, April 24-29, 1988, pp. 376-381.

[45] Lipkin H., Torfason L.E., and Duffy J.,"Efficient Motion Planning for a Planar Manipulator Based on Dexterity and Workspace Geometry," Conference on Mechanisms and Machinery, Cranfield Institute of Technology, Cranfield, United Kingdom, September 1985.

[46] Shieh M.D. and Duffy J., "Autonomous Rectilinear Motion Planning--Part I: The Geometry of the Wrist Workspace with A Single Circular Obstacle," First National Applied Mechanism and Robotics Conference, Cincinnati, Ohio, 89AMR-3A-3, Volume I, Nov. 5-8, 1989.

[47] Shieh M.D. and Duffy J., "Autonomous Rectilinear Motion Planning--Part II: The Geometry of the End-Effector Workspace and Determination of a Free Path," First National Applied Mechanism and Robotics Conference, Cincinnati, Ohio, 89AMR-3A-4, Volume I, Nov. 5-8, 1989.

[48] Shieh M.D. and Duffy J., "Autonomous Rectilinear Motion Planning--Part III: Determination of Collision Free Paths for a Planar 3R Manipulator with Multiple Circular Obstacles inside the Workspace," The Third Conference on Recent Advances in Robotics, Boca Raton, FL, May 31 -June 1, 1990.

[49] Duffy J., "Planar Mechanisms and Manipulators," Analysis of Mechanisms and Robot Manipulators, Wiley, New York, 1980, pp. 5-55.

[50] Hunt K. H., "Three-Dimensional Geometry and Spatial Mechanism," Kinematic Geometry of Mechanisms, Oxford University Press, London, 1978, pp. 246-273.

BIOGRAPHICAL SKETCH

Menq-Dar Shieh was born on Sept. 22, 1959, in Tainan, Taiwan, R.O.C. He received the Bachelor of Science in mechanical engineering from the National Chung-Hsing University in June 1981. From 1981 to 1983, he joined the Army for R.O.T.C. Service. He began graduate studies in mechanical engineering at the University of Florida in August, 1984, and received the Master of Science degree in December 1986.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph Duffy

Joseph Duffy, Chairman
Graduate Research Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Ali Seireg

Ali A. Seireg
Ebaugh Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Carl D. Crane III
Assistant Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

John Staudhammer

John Staudhammer
Professor of
Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Ralph G. Selfridge
Professor of Computer Science
and Information

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1990

Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School